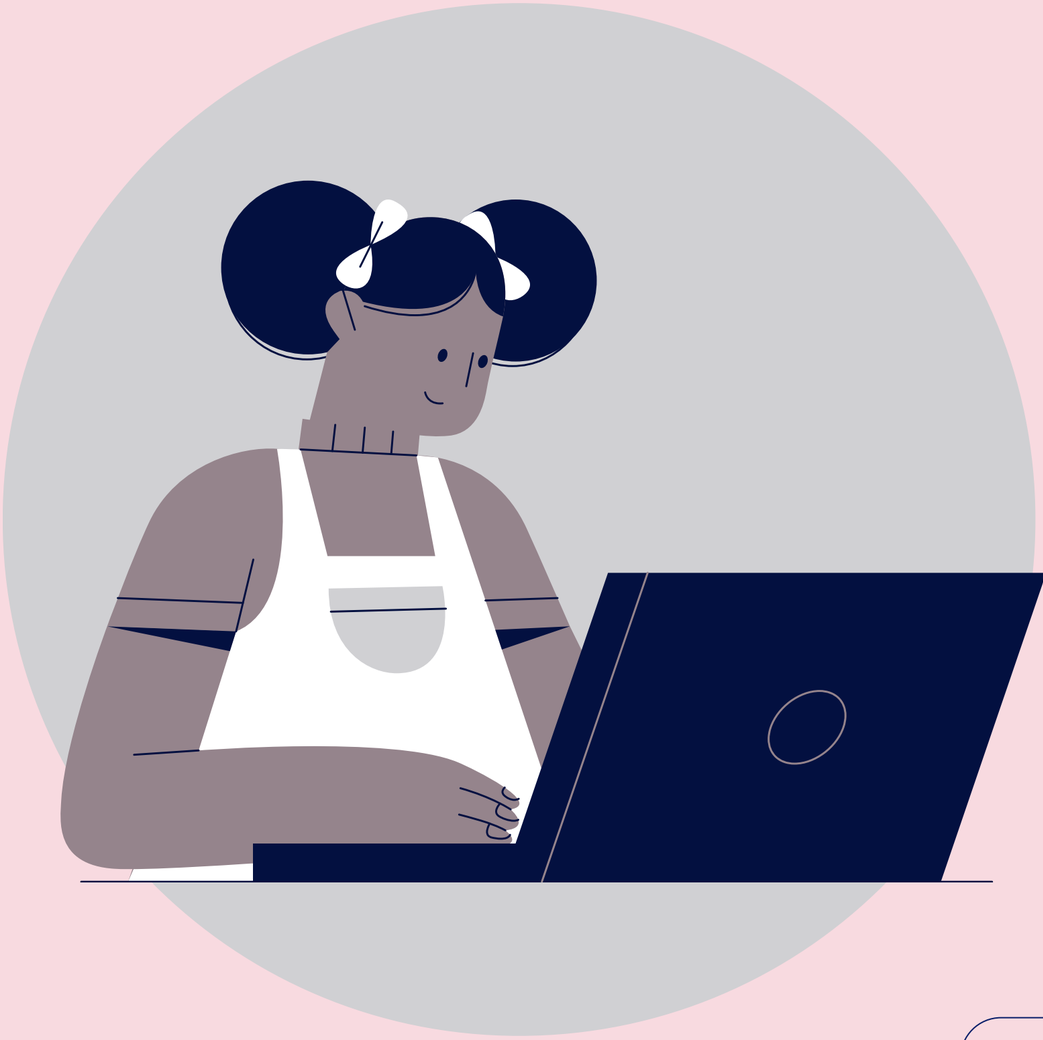
Prediction of Mental Health with Natural Language Processing

Submitted to :
Professor Iftikar Ahmad

DSC-C-09
Sepideh Azadian 56486941
Ayse Nur Safak 36061026
Yidi Zhao 26281207

CONTENTS



Problem Statement

Key Question: Predicting mental health conditions based on textual statements.

- **Objective:** Develop an AI model that classifies text into different mental health categories (e.g., Anxiety, Depression, Stress) using NLP techniques.
- **Significance:** Early detection of mental health issues can enable timely intervention and support.

Challenges

- Statements can be short and may include irrelevant information.
- Selecting an effective text representation method.
- Identifying and extracting the most relevant features for different models.
- Optimizing model performance for high accuracy.

Data Set

Data Overview:
The dataset comprises textual statements labeled with one of seven mental health statuses:

- Normal
- Depression
- Suicidal
- Anxiety
- Stress
- Bi-Polar
- Personality Disorder

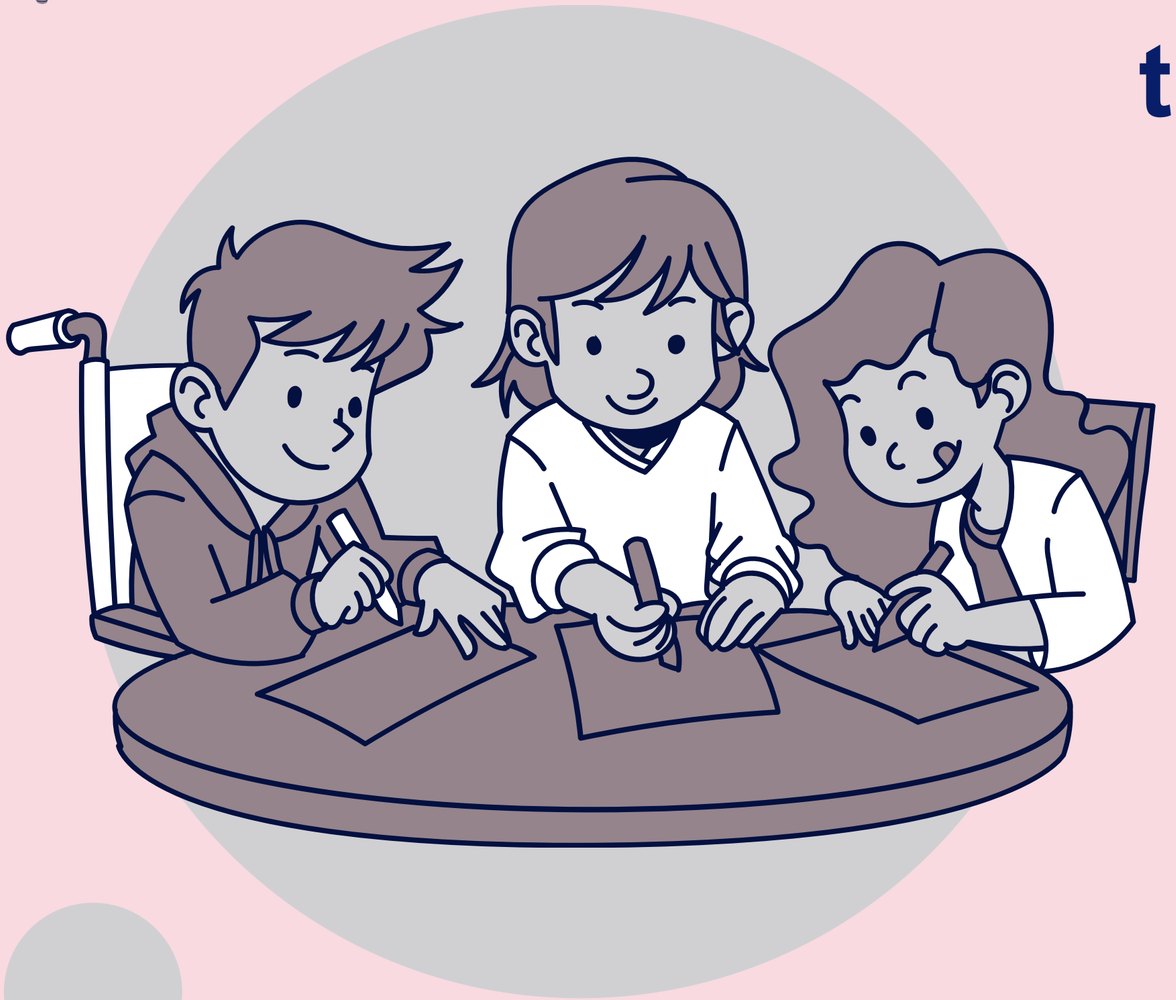
	Unnamed: 0	statement	status
0	0	oh my gosh	Anxiety
1	1	trouble sleeping, confused mind, restless hear...	Anxiety
2	2	All wrong, back off dear, forward doubt. Stay ...	Anxiety
3	3	I've shifted my focus to something else but I'...	Anxiety
4	4	I'm restless and restless, it's been a month n...	Anxiety
...	...	...	...
53038	53038	Nobody takes me seriously I’ ve (24M) dealt wit...	Anxiety
53039	53039	selfishness "I don't feel very good, it's lik...	Anxiety
53040	53040	Is there any way to sleep better? I can't slee...	Anxiety
53041	53041	Public speaking tips? Hi, all. I have to give ...	Anxiety
53042	53042	I have really bad door anxiety! It's not about...	Anxiety

Columns:

- **unique_id:** A unique identifier for each entry.
- **Statement:** The text content for analysis.
- **Mental Health Status:** The labeled classification category.

Approach to Solving the Problem

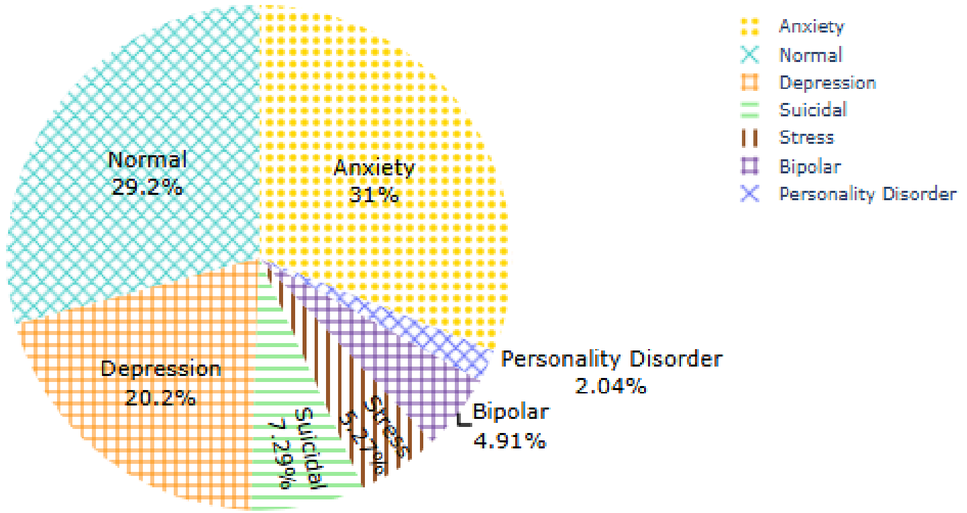
05

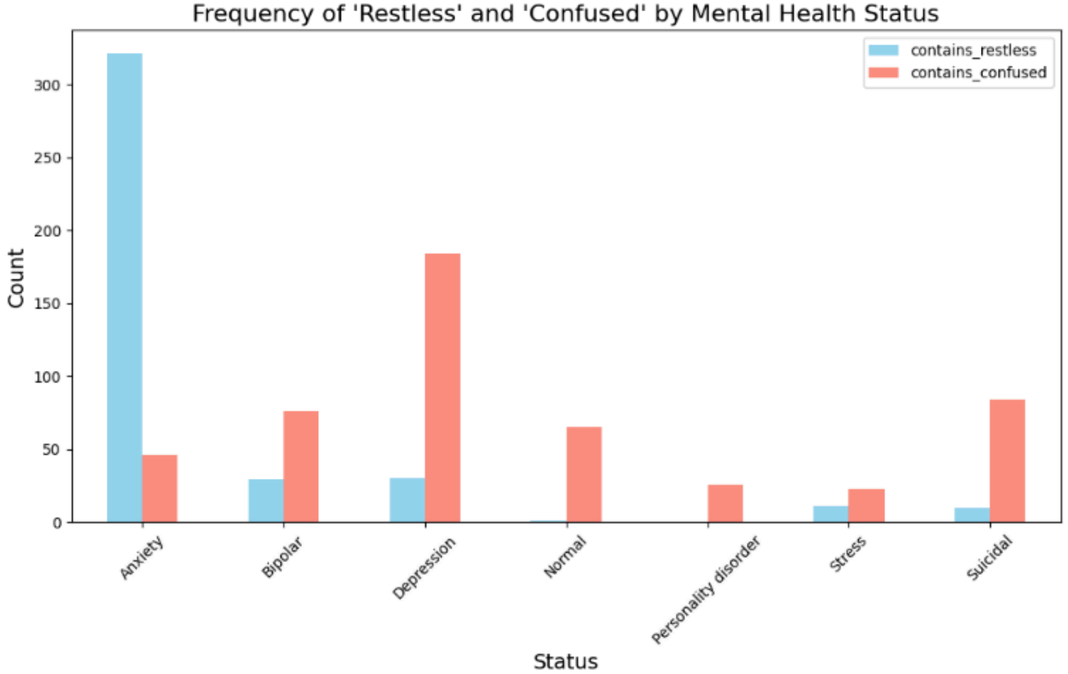


Data Preprocessing

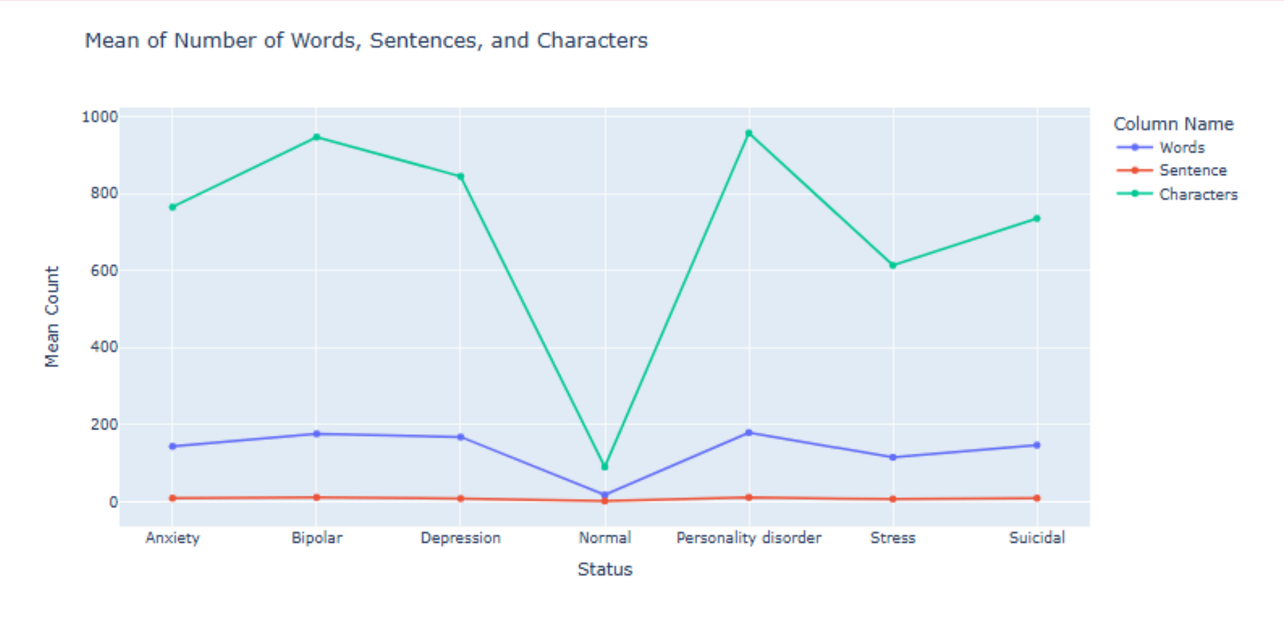
- **Text Cleaning:** Removing stop words, special characters, and handling missing data.
- **Tokenization & Vectorization:** Converting text into numerical representations using techniques like TF-IDF and BERT embeddings.
- **Feature Engineering:** Adding relevant linguistic and semantic features to improve model performance.

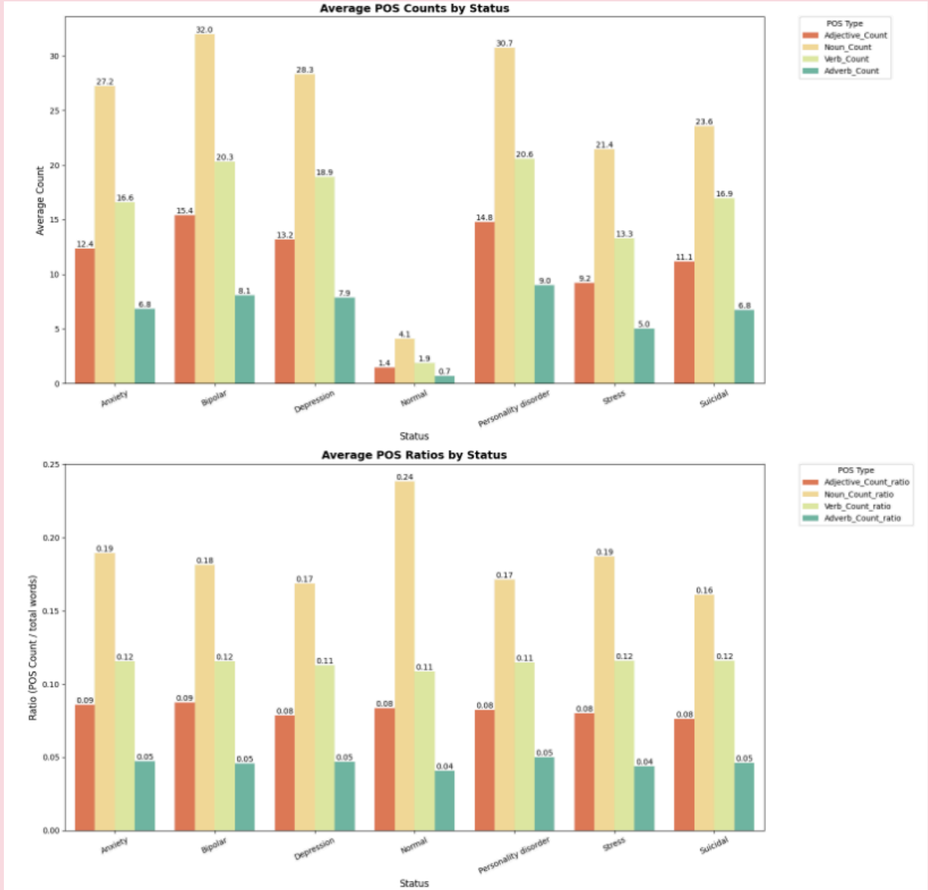
EDA


Normal
29.2%
Anxiety
31%
Depression
20.2%
Personality Disorder
2.04%
Bipolar
4.91%
Suicidal
7.29%
Stress
5.21%
Anxiety
Normal
Depression
Suicidal
Stress
Bipolar
Personality Disorder

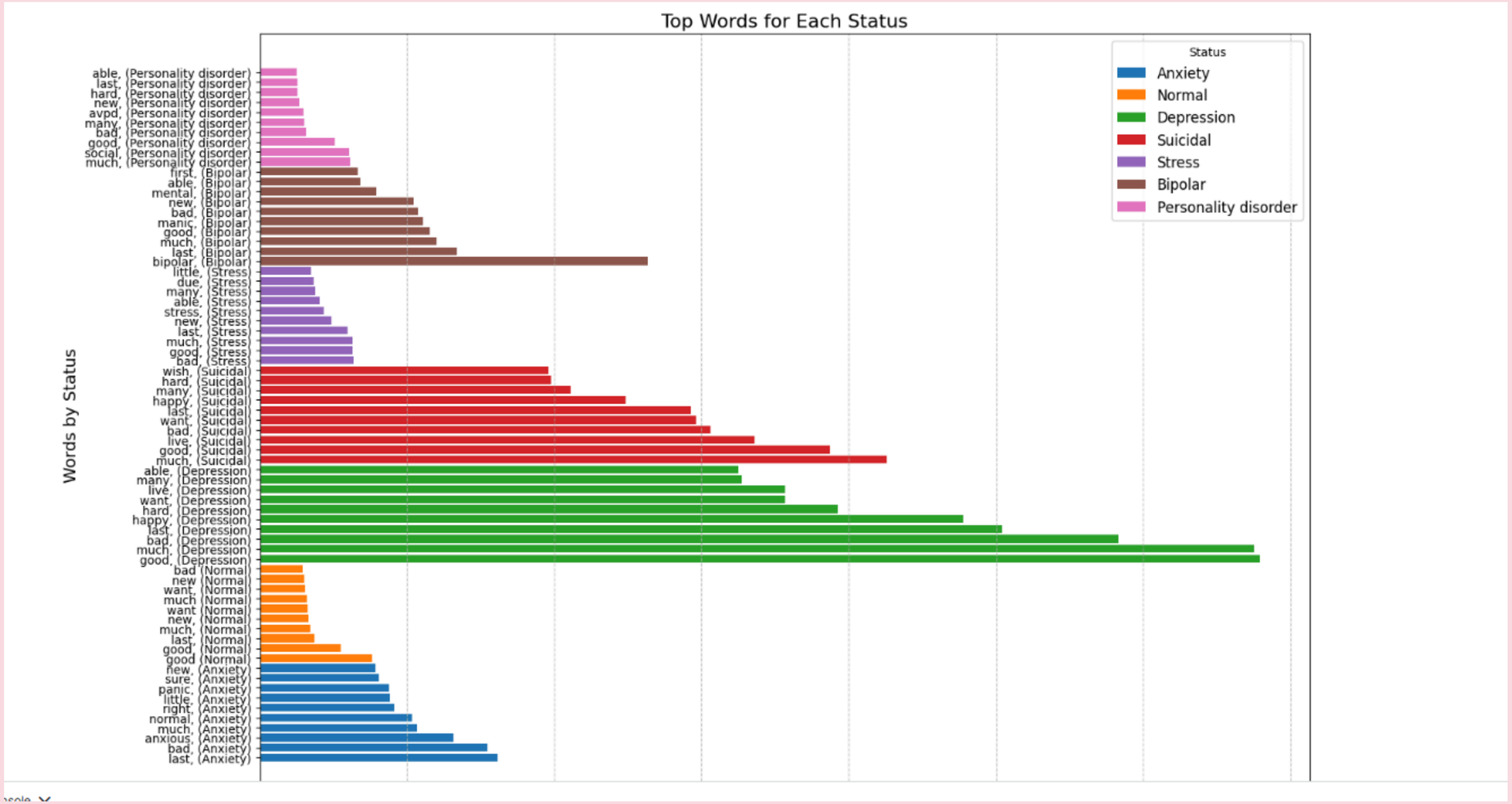

Frequency of 'Restless' and 'Confused' by Mental Health Status
contains_restless
contains_confused
Count
0
50
100
150
200
250
300
Anxiety
Bipolar
Depression
Normal
Personality disorder
Stress
Suicidal
Status

EDA


Mean of Number of Words, Sentences, and Characters
Mean Count
1000
800
600
400
200
0
Anxiety
Bipolar
Depression
Normal
Personality disorder
Stress
Suicidal
Status
Column Name
Words
Sentence
Characters


Average POS Counts by Status
Average Count
Status
POS Type
Adjective_Count
Noun_Count
Verb_Count
Adverb_Count
Average POS Ratios by Status
Ratio (POS Count / total words)
Status
POS Type
Adjective_Count_ratio
Noun_Count_ratio
Verb_Count_ratio
Adverb_Count_ratio
Anxiety
Bipolar
Depression
Normal
Personality disorder
Stress
Suicidal

EDA


Top Words for Each Status
Words by Status
Status
Anxiety
Normal
Depression
Suicidal
Stress
Bipolar
Personality disorder
able, (Personality disorder)
last, (Personality disorder)
hard, (Personality disorder)
new, (Personality disorder)
avpd, (Personality disorder)
many, (Personality disorder)
bad, (Personality disorder)
good, (Personality disorder)
social, (Personality disorder)
much, (Personality disorder)
first, (Bipolar)
able, (Bipolar)
mental, (Bipolar)
new, (Bipolar)
bad, (Bipolar)
manic, (Bipolar)
good, (Bipolar)
much, (Bipolar)
last, (Bipolar)
bipolar, (Bipolar)
little, (Stress)
due, (Stress)
many, (Stress)
able, (Stress)
stress, (Stress)
new, (Stress)
last, (Stress)
much, (Stress)
good, (Stress)
bad, (Stress)
wish, (Suicidal)
hard, (Suicidal)
many, (Suicidal)
happy, (Suicidal)
last, (Suicidal)
want, (Suicidal)
bad, (Suicidal)
live, (Suicidal)
good, (Suicidal)
much, (Suicidal)
able, (Depression)
many, (Depression)
live, (Depression)
want, (Depression)
hard, (Depression)
happy, (Depression)
last, (Depression)
bad, (Depression)
much, (Depression)
good, (Depression)
bad (Normal)
new (Normal)
want, (Normal)
much (Normal)
want (Normal)
new, (Normal)
much, (Normal)
last, (Normal)
good, (Normal)
good (Normal)
new, (Anxiety)
sure, (Anxiety)
panic, (Anxiety)
little, (Anxiety)
right, (Anxiety)
normal, (Anxiety)
much, (Anxiety)
anxious, (Anxiety)
bad, (Anxiety)
last, (Anxiety)

Model Selection

Experiment with 1 baseline model and 4 advanced models:

1. Baseline Model: Logistic Regression
2. Advanced Models:
   - Feedforward Neural Network (FNN)
   - Multi-layer Perceptron (MLP)
   - Long Short-Term Memory (LSTM)
   - XgBoots+BERT

1. **Logistic Regression (Baseline):** A simple linear model that predicts mental health conditions based on word frequency and statistical relationships.
2. **Feedforward Neural Network (FNN):** A basic neural network that captures non-linear relationships in the text data.
3. **Multi-layer Perceptron (MLP):** A deep learning model with multiple hidden layers that improves classification by learning complex patterns.
4. **Long Short-Term Memory (LSTM):** A recurrent neural network (RNN) variant that captures contextual meaning and dependencies in text over long sequences.
5. **BERT + XGBoost for Text Classification :** performs text classification using BERT embeddings and XGBoost while handling class imbalance through resampling.

Logistic Regression (Baseline):

- Assumes a linear relationship between word frequency and class labels.
- Uses TF-IDF for feature extraction.
- Key Parameters: Regularization (L1/L2), C (inverse of regularization strength).

Feedforward Neural Network (FNN):

- A simple neural network with fully connected layers.
- Extracts non-linear relationships between features.
- Key Parameters: Number of layers, activation functions, learning rate, dropout.

Multi-layer Perceptron (MLP):

- A deeper version of FNN with multiple hidden layers.
- Uses backpropagation and optimizers like Adam or SGD.
- Key Parameters: Number of hidden layers, neurons per layer.

Long Short-Term Memory (LSTM):

- A recurrent neural network (RNN) variant, effective for sequential data.
- Captures long-term dependencies in text sequences.
- Key Parameters: Number of LSTM units, dropout rate, optimizer, sequence length.

BERT + XGBoost

- Sampling & Balancing the Data (to avoid class imbalance issues)
- Embedding Text using BERT
  - (Bidirectional Encoder Representations from Transformers is a deep learning model developed by Google that:
  - Understands contextual meaning by looking at words in both directions (left and right).
- Training an XGBoost Classifier (XGBoost (Extreme Gradient Boosting) is one of the most powerful tree-based machine learning models.)

Evaluation Criteria

To measure the effectiveness of the models, we use the following metrics:

- **Accuracy:** The percentage of correctly classified instances.
- **Precision & Recall:** To evaluate the balance between false positives and false negatives.
- **F1-Score:** A harmonic mean of precision and recall.
- **Support:** Number of true instances per class.

Results(Logistic Regression)


```
lr=LogisticRegression(random_state=42, max_iter=5000)
lr_params={'penalty': ['l1', 'l2','elasticnet'],'C':[0.01, 0.1, 1, 10, 100]}
lr_grid=GridSearchCV(lr,lr_params,cv=3,scoring='accuracy')
lr_grid.fit(X_train,y_train_le)
print("Best Logistic Regression Params:", lr_grid.best_params_)
```

```
Best Logistic Regression Params: {'C': 1, 'penalty': 'l2'}
```

+ Code + Markdown

```
best_lr_model=lr_grid.best_estimator_
```

90]:
```
best_lr_model
```

90]:
```
LogisticRegression(C=1, max_iter=5000, random_state=42)
```

92]:
```
accuracy_lr=accuracy_score(y_test_le,y_pred_lr)
```

93]:
```
print(f"Accuracy of the best Logistic Regression model: {accuracy_lr * 100:.2f}%")
```

```
Accuracy of the best Logistic Regression model: 74.83%
```

94]:
```
print("Classification Report for Logistic Regression:")
print(classification_report(y_test_le, y_pred_lr))
```

```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       0.82      0.71      0.76       768
           1       0.89      0.62      0.73       556
           2       0.68      0.74      0.71      3081
           3       0.83      0.95      0.88      3269
           4       0.89      0.40      0.55       215
           5       0.68      0.37      0.47       517
           6       0.67      0.62      0.64      2131

    accuracy                           0.75     10537
   macro avg       0.78      0.63      0.68     10537
weighted avg       0.75      0.75      0.74     10537
```

Results(FNN)


```
Epoch 1/10
527/527 [==============================] - 5s 9ms/step - loss: 1.1493 - accuracy: 0.5853 - val_loss: 0.7826 - val_accuracy: 0.7191
Epoch 2/10
527/527 [==============================] - 4s 7ms/step - loss: 0.7194 - accuracy: 0.7406 - val_loss: 0.6887 - val_accuracy: 0.7494
Epoch 3/10
527/527 [==============================] - 4s 8ms/step - loss: 0.5859 - accuracy: 0.7907 - val_loss: 0.6743 - val_accuracy: 0.7545
Epoch 4/10
527/527 [==============================] - 4s 8ms/step - loss: 0.5058 - accuracy: 0.8211 - val_loss: 0.6795 - val_accuracy: 0.7543
Epoch 5/10
527/527 [==============================] - 4s 7ms/step - loss: 0.4368 - accuracy: 0.8458 - val_loss: 0.7010 - val_accuracy: 0.7530
Epoch 6/10
527/527 [==============================] - 4s 7ms/step - loss: 0.3839 - accuracy: 0.8668 - val_loss: 0.7260 - val_accuracy: 0.7512
Epoch 7/10
527/527 [==============================] - 4s 7ms/step - loss: 0.3320 - accuracy: 0.8850 - val_loss: 0.7684 - val_accuracy: 0.7483
Epoch 8/10
527/527 [==============================] - 4s 7ms/step - loss: 0.2871 - accuracy: 0.9032 - val_loss: 0.8114 - val_accuracy: 0.7487
Epoch 9/10
527/527 [==============================] - 4s 7ms/step - loss: 0.2456 - accuracy: 0.9193 - val_loss: 0.8551 - val_accuracy: 0.7466
Epoch 10/10
527/527 [==============================] - 4s 7ms/step - loss: 0.2089 - accuracy: 0.9339 - val_loss: 0.8936 - val_accuracy: 0.7449
Test Accuracy: 0.7395
                     precision    recall  f1-score   support

            Anxiety       0.77      0.76      0.76       768
            Bipolar       0.78      0.73      0.75       556
         Depression       0.69      0.67      0.68      3081
             Normal       0.87      0.91      0.89      3269
Personality disorder       0.69      0.55      0.61       215
             Stress       0.58      0.48      0.53       517
           Suicidal       0.62      0.65      0.64      2131

           accuracy                           0.74     10537
          macro avg       0.72      0.68      0.70     10537
       weighted avg       0.74      0.74      0.74     10537
```

Results(MLP)

Neural Network Results:

	precision	recall	f1-score	support
0	0.80	0.73	0.76	768
1	0.76	0.72	0.74	556
2	0.65	0.72	0.68	3081
3	0.87	0.91	0.89	3269
4	0.64	0.62	0.63	215
5	0.57	0.49	0.52	517
6	0.64	0.55	0.59	2131
accuracy			0.73	10537
macro avg	0.70	0.68	0.69	10537
weighted avg	0.73	0.73	0.73	10537

Accuracy: 0.7310429913637658

+ Code + Markdown

Results(LSTM)

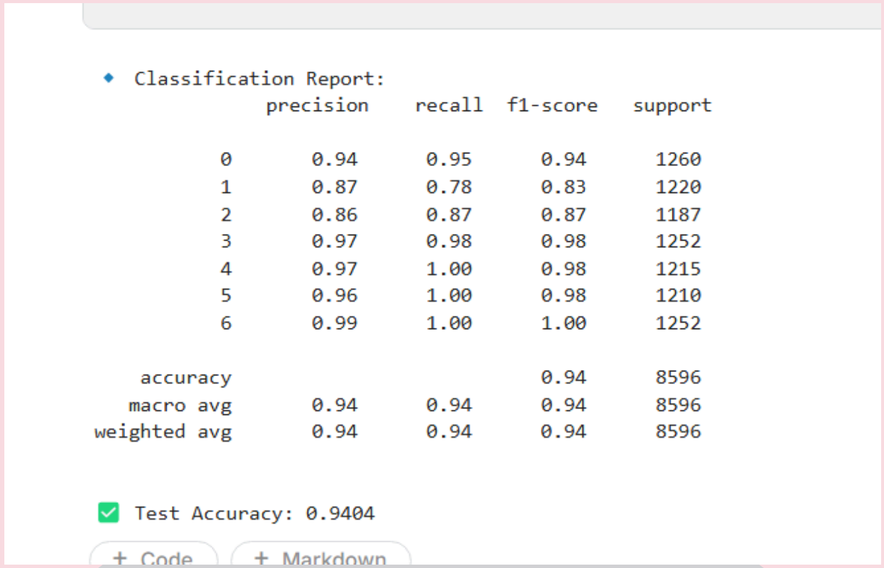

```
Epoch 1/5
1317/1317 [==============================] - 96s 71ms/step - loss: 1.1606 - accuracy: 0.5583 - val_loss: 0.9571 - val
_accuracy: 0.6082
Epoch 2/5
1317/1317 [==============================] - 94s 71ms/step - loss: 0.9162 - accuracy: 0.6569 - val_loss: 0.8823 - val
_accuracy: 0.6750
Epoch 3/5
1317/1317 [==============================] - 101s 76ms/step - loss: 0.7880 - accuracy: 0.7186 - val_loss: 0.8032 - va
l_accuracy: 0.7170
Epoch 4/5
1317/1317 [==============================] - 95s 72ms/step - loss: 0.6856 - accuracy: 0.7594 - val_loss: 0.7677 - val
_accuracy: 0.7253
Epoch 5/5
1317/1317 [==============================] - 95s 72ms/step - loss: 0.6069 - accuracy: 0.7884 - val_loss: 0.7622 - val
_accuracy: 0.7295
330/330 [==============================] - 9s 25ms/step
LSTM Results:
              precision    recall  f1-score   support

           0       0.65      0.80      0.72       768
           1       0.77      0.69      0.73       556
           2       0.70      0.61      0.66      3081
           3       0.90      0.90      0.90      3269
           4       0.75      0.33      0.45       215
           5       0.44      0.49      0.47       517
           6       0.62      0.73      0.67      2131

    accuracy                           0.73     10537
   macro avg       0.69      0.65      0.66     10537
weighted avg       0.74      0.73      0.73     10537
```

Results(BERT+XGBOOST

- models correctly predict 94 percent of test samples
-



🔹 Classification Report:

	precision	recall	f1-score	support
0	0.94	0.95	0.94	1260
1	0.87	0.78	0.83	1220
2	0.86	0.87	0.87	1187
3	0.97	0.98	0.98	1252
4	0.97	1.00	0.98	1215
5	0.96	1.00	0.98	1210
6	0.99	1.00	1.00	1252
accuracy			0.94	8596
macro avg	0.94	0.94	0.94	8596
weighted avg	0.94	0.94	0.94	8596

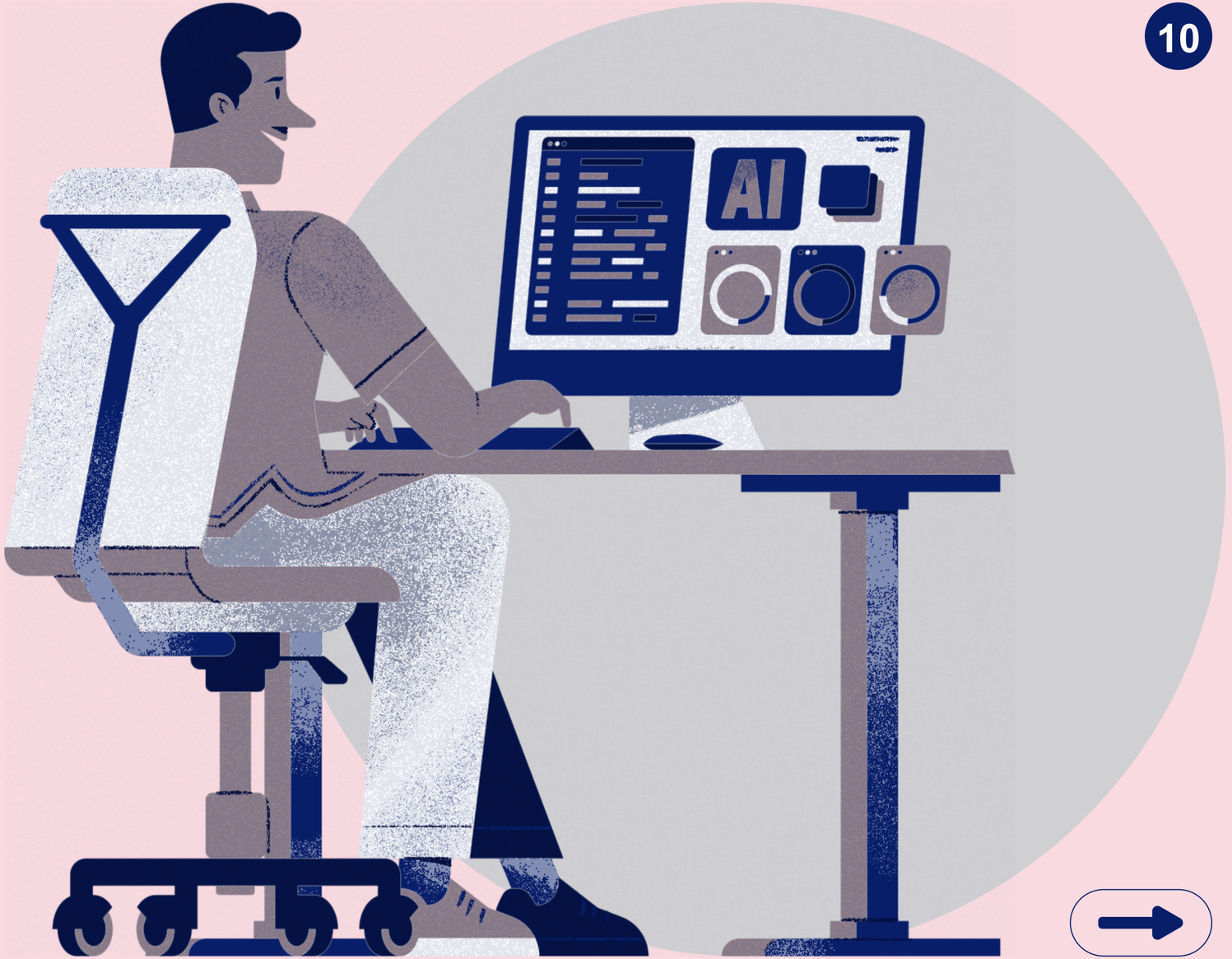
✅ Test Accuracy: 0.9404

+ Code + Markdown

Conclusion: The use of learning models like XGBoost generally yields higher accuracy in classifying mental health conditions compared to traditional approaches. However, further improvements can be made with better feature selection and fine-tuning of hyperparameters.

PAUCEK
AND LAGE

THANK YOU



REFERENCES

DATASET:

https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health