LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC6057NI Applied Machine Learning

Submission: Final Submission

AY 2025 2026

Academic Semester: Autumn Semester 2025

Student Name: Sabin Adhikari

London Met ID: 23048604

College ID: NP01AI4A230033

Assignment Due Date: Monday, January 19, 2025

Assignment Submission Date: Monday, January 19, 2025

Github Link	https://github.com/SabinAdhikarii/ML-Project-Mental-Health-Sentiment-Analysis

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded

Contents

Table of Figures

Table

1. INTRODUCTION

1.1 Project Overview

The rapid curative development of digital textual information on social media sites, Internet forums, blogs, and support groups in the past decade has presented an opportunity and a challenge when it comes to studying human feelings and behavioral patterns. Such data can only be handled in large quantities using the tools of Artificial Intelligence (AI) and Natural language Processing (NLP) due to the impossibility of traditional manual approaches of categorizing and interpreting them. Text classification, which is one of the most important tasks in NLP, allows automatically assigning pre-established labels to the written materials depending on their semantic content. This is a project, which uses supervised machine learning and deep learning models to categorize mental health conditions using text data.

The main objective of the project is to design, develop, and test an end-to-end pipeline to classify the mental health conditions. The pipeline combines text preprocessing, feature extraction and model training through classical machine learning algorithms.

1.2 Introduction to Machine Learning and NLP

1.2.1 Concepts of Machine Learning Used

Machine Learning (ML) is a branch of AI that involves creating algorithms that can learn trends on the data and provide predictions based on unknown inputs. The project involves the use of supervised learning, in which one or another text sample is linked to the known label (mental health condition). The classification target is a multi-class task wherein the model is expected to make predictions in one of the various available categories.

The machine learning models applied in the project are mentioned below:

- Logistic Regression
- Random Forest
- Multinomial Naive Bayes

1.2.2 Concepts of Natural Language Processing Used

NLP offers the basis of converting raw text to a structured format that can be used by ML models. Some of the important concepts used are:

- **Text Preprocessing**: lowercasing, punctuations, tokenization and stop-word removal and lemmatization.
- **Feature Extraction**: TF-IDF to emphasize words of discrimination and contextual embeddings.
- **Modeling**: In this stage, different classical machine learning models like Logistic Regression, Random Forest and Naïve Bayes is implemented. Each model learns the pattern in features to classify the text.
- **Evaluation Metrics**: The 3 models used are evaluated with the classification report for accuracy, precision, recall and f1-score. ROC curve, confusion matrices are analyzed.
- **Model Comparison**: The output of the all three model are compared to check which model is performing best correctly identifying the true positive and false negative.

2. PROBLEM DOMAIN

2.1 Text Classification Problem

The text classification is a fundamental logic of NLP where a predefine class is assigned to textual data. The problem of text classification problem is to analyze the text samples and predicts the corresponding labels. It is required to transform raw text data into suitable form that models can process. The main issue this work discussed is that of automated categorization of mental-health issues on the basis of text nature. People often use digital media like social-media posts, online forums, blogs and support communities to express their affective and psychological mind. Even though they are unstructured, noisy, and informal, they are analogous to ciphers and carry useful information concerning mental-health conditions.

Mathematically the problem can be stated as a multi-class, single-label classification problem: learn a mapping (also known as an extractor)

$$f: D \rightarrow C,$$

where f is used on a document, D is a collection of texts, C is the collection of diagnostic classes (depression, anxiety, bipolar disorder, sucidal, happy). Each text belongs to a single class only.

The problems that can occur in this field are:

- **Diminished vocabulary:** Incorrectly used words like tired or stressed can be related to several diseases.
- **Noise and informality:** Social-media texts are often full of abbreviations and misspellings, slang articles.
- **Ethical sensitivity:** It is also possible to make mistakes with misclassification in non- research situations (Bentley et al., 2020).

2.2 SELECTION OF DATASET

2.2.1 Criteria for Dataset Selection

For a project focused on mental health condition classification, the dataset must meet several critical criteria:

- **Textual content**: The dataset should contain raw text samples that reflect real- world expressions of emotions and psychological states.
- **Labeled categories**: Each text must be associated with a condition label (e.g., depression, anxiety, happy), enabling supervised learning.
- **Scale and diversity**: A sufficiently large dataset is required to train both classical machine learning models and deep learning architectures.
- **Balance**: Classes should be reasonably balanced to avoid bias in predictions.
- **Accessibility and licensing**: The dataset should be openly available for academic use, with clear licensing terms.

2.2.2 Dataset Chosen

Based on these criteria, we selected the Mental Health Condition Classification Dataset hosted on Hugging Face by contributor sai1908 (Hugging Face Dataset Card).

Characteristic	Value/Description
Dataset Name	Mental Health Condition Classification
Source	HuggingFace (sai1908/Mental_Health_Condition_Classification) https://huggingface.co/datasets/sai1908/Mental_Health_Condition_Classification
Total Samples	103,488
Number of Classes	7
Data Type	Text (mental health descriptions)
Average Text Length	82 words
Minimum Text Length	1 word
Maximum Text Length	5,419 words

Table 1: Dataset Overview

2.3 History or Background of Dataset

The dataset used in this study is the Mental Health Condition Classification Dataset that is currently available on Hugging Face by its owner sai1908. It consists of around 103,000 textual samples that were allotted to a number of conditions, such as depression, anxiety, bipolar and PTSD, OCD, schizophrenia, ADHD, eating conditions, and positive affect. Hugging Face provides a unified interface of accessing datasets and experimenting; however, the main data are open-source corpora of materials about mental-health-related issues collected at social media, online discussion forums, and support communities.

Before it was posted on Hugging Face, computational psychiatry researchers and affective computing already maintained collections of mental-health discourse, frequently derived as Twitter posts, Reddit forums like r/depression and r/anxiety, and career support forums where people can discuss themselves. As an example, De Choudhury et al. (2013) were the first to utilise Twitter data to forecast depressive symptoms, and Coppersmith et al. (2015) generated annotated tweet corpora of mental health indicators.

These were the first steps to larger, multi-class datasets currently available through Hugging Face (Calvo et al., 2017).

2.4 Societal and Business Relevance of Text Classification

2.4.1 Societal Relevance of Text Classification

Mental health text classification has significant societal relevance as automatically classify the textual contents into categories like depression, sad, happy, anxiety, etc. In the area of mental health monitoring, sentiment analysis of social media post, comments, message, etc could be significantly beneficial for tracking and identifying the people suffering from such conditions. Not only this but it could be equally useful to detect harmful or abusive contents online, and filtering out them from social media platforms.

2.4.2 Business Relevance of Text Classification

From the business view, mental health classification is very important and impactful to analyze the customer behavior, their sentiments, reviews and support tickets which could significantly increases the customer satisfaction from the platform. The business could also earn goodwill and wild reputation by saving the life of people suffering from different disorders. By automating the categorization of text, organization can save time, reduce cost, gain competitive advantages, goodwill from data driven decision making.

3. PROJECT AS SOLUTION

The mental health text classification system is important as it could save the individual suffering from different disorders. The proposed system here is the NLP based machine learning project that automatically classify the text into best fit category. The system translates raw and noisy text into organized forms which can then be classified.

3.1 Text Preprocessing

- Lowercasing
- Deletion of URLs, mentions, hashtags, numeric tokens, and punctuations.
- Tokenization
- Stop-word removal
- Lemmatization

These are the processes that normalize the text and minimize noise so that they are consistent with what the best practices of natural-language processing stipulate (Manning et al., 2008).

3.2 Feature Extraction

- TFIDF, as predictive of discriminative words among documents (Salton and Buckley, 1988).
- Contextual embeddings revealed by model, containing stronger semantic and contextual knowledge (Devlin et al., 2019).

3.3 Model Training

TF-IDF features were used to train classical machine-learning models, such as Logistic regression, random forest and naive bayes. The logistic regression is served as baseline model where random forest model and naive bayes is analyze on basis of logistic model. The 80% of the data is allotted for training purpose while remaining 20% for testing purpose.

3.4 Model Evaluation

Accuracy, precision, recall, F1 -score, confusion matrixes and macro/weighted averages were used to evaluate model performance by considering class imbalance. The classification report of all three model are evaluated and the best fit model is chosen to classify the mental health text classification. To view the model in action, sample text is taken and observed if model predicts the correctly or not.

4. Machine learning models that were used.

4.1 Logistic Regression

A linear discriminative classifier that is biased towards high-dimensional and sparse data as TF-IDF vectors. It provides interpretability based on the feature-weight analysis (Bishop, 2006).

For K classes:

$$P(y = k \mid x) = \frac{e^{w_k^T x}}{\sum_{j=1}^{K} e^{w_j^T x}}$$

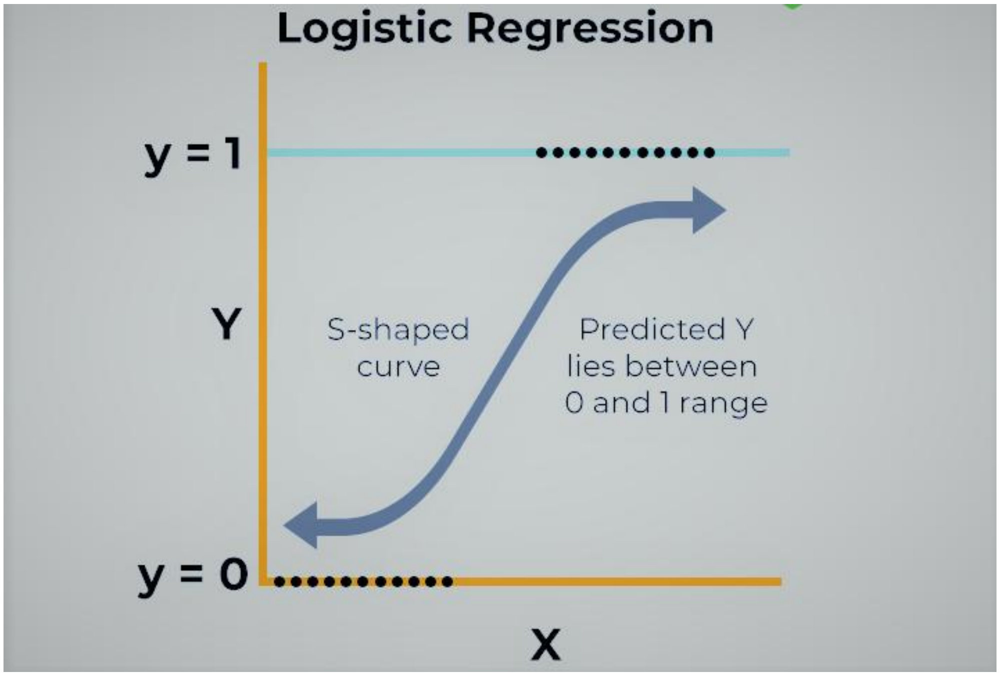


Figure 1: Logistic Regression

4.2 Random Forest

A combination of decision trees that enhances robustness and a decision tree that is useful in non-linear relationships thus representing interaction between complex features (Breiman, 2001).

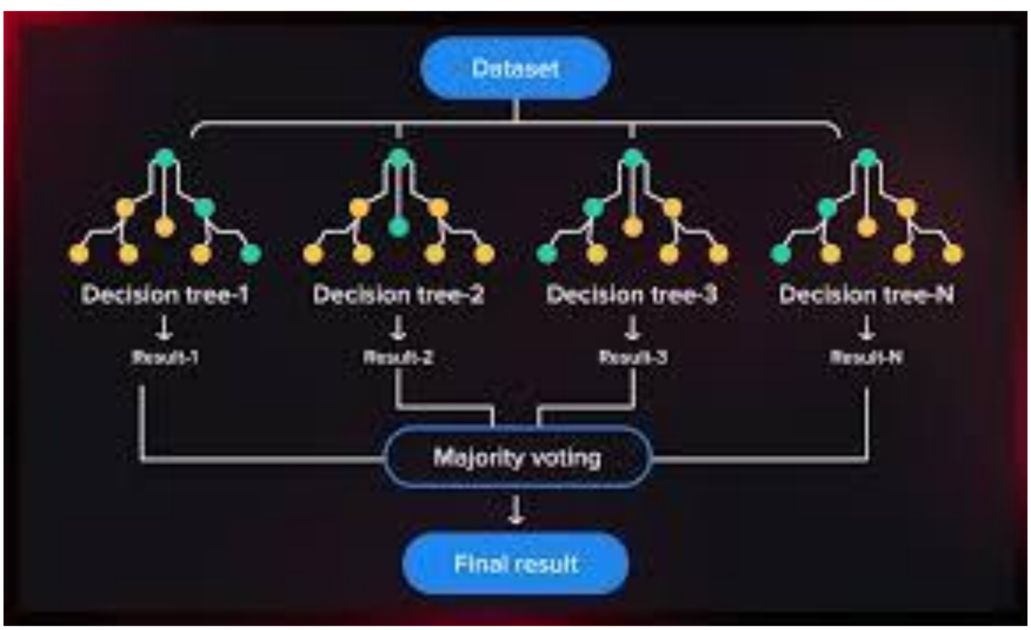


Figure 2: Random Forest

4.3 Multinomial Naive Bayes

This is a machine learning algorithm that models the frequency of words as a count and assume all words is multinomially distributed.

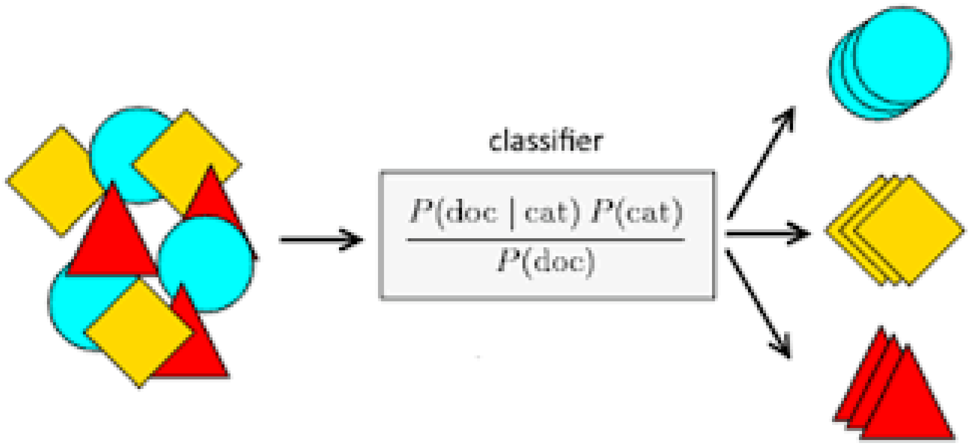


Figure 3: Multinomial Naive Bayes

5. Pseudocode for Mental Health Condition Classification Project

BEGIN

LOAD dataset

DISPLAY dataset statistics (total samples, classes, distribution)

GENERATE visualizations (word clouds, class distribution charts)

FOR each text sample IN dataset:

CONVERT text to lowercase

REMOVE special characters and digits

TOKENIZE text into words

REMOVE stop words

APPLY lemmatization to words

JOIN processed words back into text

END FOR

```
SPLIT dataset into training (80%) and testing (20%) set

TRANSFORM training text to TF-IDF features
TRANSFORM testing text to TF-IDF features

INITIALIZE Logistic Regression model
TRAIN Logistic Regression on training features and labels
PREDICT on test set

INITIALIZE Random Forest
TRAIN Random Forest on training features and labels
PREDICT on test set

INITIALIZE Multinomial Naive Bayes model
TRAIN Naive Bayes on training features and labels
PREDICT on test set

FOR each model (Logistic Regression, Random Forest, Naive Bayes):
    CALCULATE accuracy score, precision, recall, F1-score
    GENERATE classification report for all classes
    COMPUTE confusion matrix
    CALCULATE ROC-AUC for each class
    STORE all metrics
END FOR

PLOT confusion matrices, roc curve for all models

CREATE comparison table with all metrics
DISPLAY performance of each model side-by-side
```

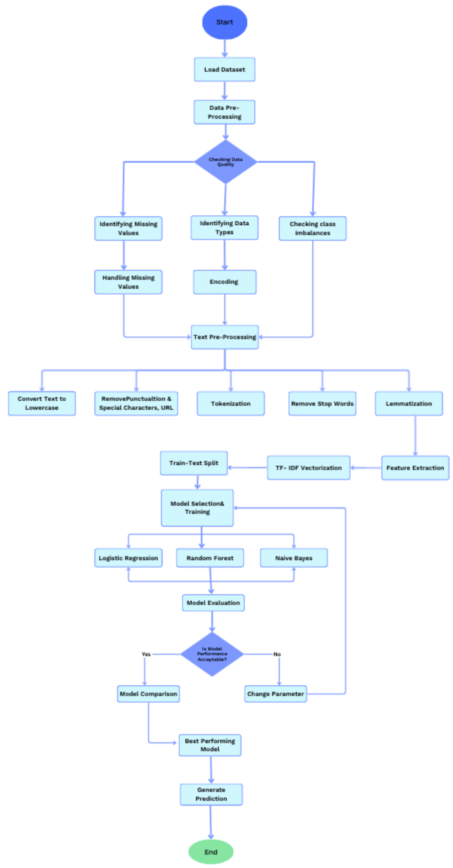
```
    IDENTIFY best performing model based on F1-score
    ANALYZE class-specific performance differences


FOR each sample text:
        PREPROCESS text using same pipeline
        TRANSFORM to TF-IDF features
        PREDICT using all three models
        DISPLAY predicted class and confidence for each model
    END FOR
```

6. System Flowchart and System Architecture

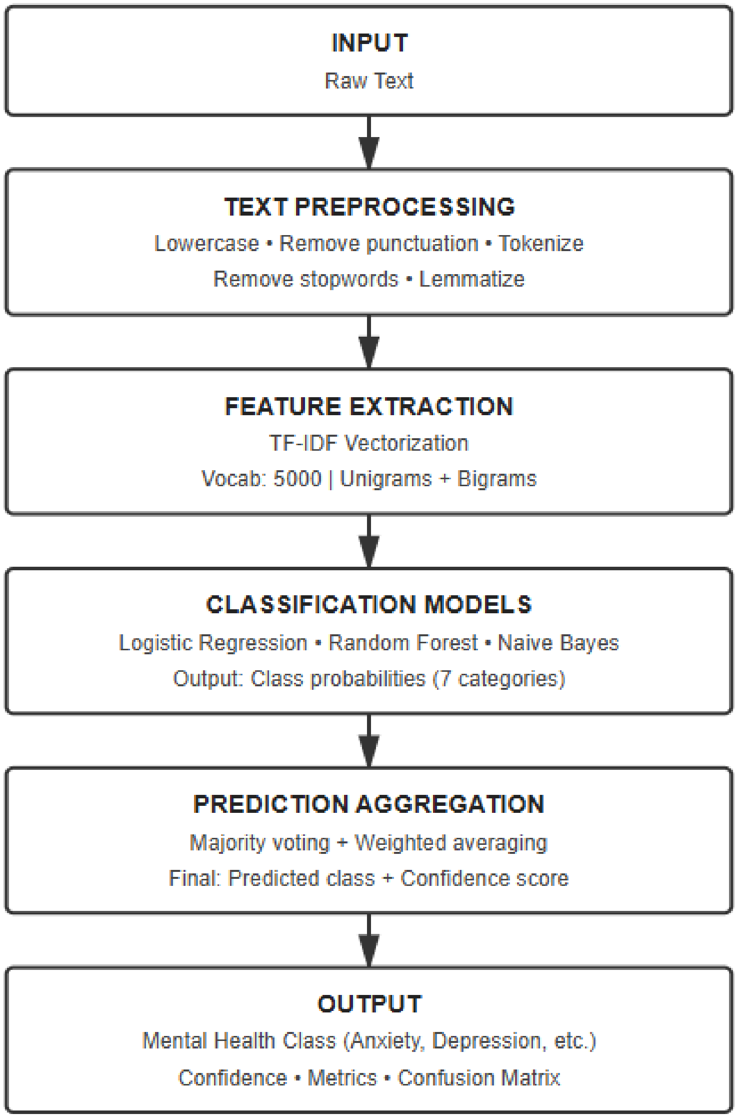
6.1 System Flowchart

Figure 4: System Flowchart



6.2 System Architecture

Figure 5: System Architecture



7. Project Development

The proposed system Mental Health Text Classification system is developed following NLP and machine learning algorithms. The dataset is loaded to the machine first then to get a detail information about the dataset, its shape, size, and columns. After detail understanding of the dataset EDA is done to visually understand the data. Text Pre processing is done using NLP concepts like TF-IDF, and Vectorization. The processed text data is then splitted into train test percentage and the machine learning models Logistic Regression, Naïve Bayes and Random Forest. The classification report of all model is compared and best model is choosed to predict the textual data.

7.1 Tools and Tech Stack Used

7.1.1 Python Programming Language

For the development of the project, python has been used as a programming language because of its platform friendliness and wide spread models and libraries.

Figure 6: Python

7.1.2 Pandas and Numpy for Data Manipulation

For the data manipulation, panda and numpy libraries of python is used.

Figure 7: Pandas and Numpy for Data Manipulation

7.1.3 Sci-kit Learn

Sci-learn is used in the project for model training, and evaluation purpose.

Figure 8: Sci-kit Learn

8. Result

8.1 Model Evaluation Result of Logistic Regression

4.3 Logistic Regression Evaluation

```
# Make predictions
y_pred_lr = lr_model.predict(X_test_tfidf)
y_pred_lr_proba = lr_model.predict_proba(X_test_tfidf)

lr_accuracy = accuracy_score(y_test, y_pred_lr)
lr_precision = precision_score(y_test, y_pred_lr, average='weighted')
lr_recall = recall_score(y_test, y_pred_lr, average='weighted')
lr_f1 = f1_score(y_test, y_pred_lr, average='weighted')

# Detailed classification report
print("DETAILED CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred_lr, target_names=class_names, digits=4))
```

```
DETAILED CLASSIFICATION REPORT
                      precision    recall  f1-score   support

             anxiety     0.9356    0.9407    0.9382      3524
             bipolar     0.9817    0.9373    0.9590      2742
              stress     0.6791    0.7167    0.6974      3180
personality disorder     0.8411    0.9437    0.8894      3214
              normal     0.9859    0.9536    0.9695      2783
          depression     0.9524    0.8943    0.9225      3046
            suicidal     0.6639    0.6080    0.6347      2209

            accuracy                         0.8657     20698
           macro avg     0.8628    0.8563    0.8587     20698
        weighted avg     0.8679    0.8657    0.8659     20698
```

Figure 9: Screenshot of Evaluation Result of Logistic Regression

8.2 Model Evaluation Result of Random Forest

5.2 Random Forest Evaluation

```python
# Make predictions
y_pred_rf = rf_model.predict(X_test_tfidf)
y_pred_rf_proba = rf_model.predict_proba(X_test_tfidf)

# Calculate metrics
rf_accuracy = accuracy_score(y_test, y_pred_rf)
rf_precision = precision_score(y_test, y_pred_rf, average='weighted')
rf_recall = recall_score(y_test, y_pred_rf, average='weighted')
rf_f1 = f1_score(y_test, y_pred_rf, average='weighted')

print("RANDOM FOREST - PERFORMANCE METRICS")
print(f"\n Accuracy:  {rf_accuracy:.4f} ({rf_accuracy*100:.2f}%)")
print(f"Precision: {rf_precision:.4f}")
print(f"Recall:    {rf_recall:.4f}")
print(f"F1-Score:  {rf_f1:.4f}")

# Detailed classification report
print("DETAILED CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred_rf, target_names=class_names, digits=4))
```

```
RANDOM FOREST - PERFORMANCE METRICS

 Accuracy:  0.7805 (78.05%)
Precision: 0.8226
Recall:    0.7805
F1-Score:  0.7695
DETAILED CLASSIFICATION REPORT
                      precision    recall  f1-score   support

             anxiety     0.9208    0.8351    0.8759      3524
             bipolar     0.9905    0.8778    0.9308      2742
              stress     0.5484    0.7387    0.6295      3180
personality disorder     0.5927    0.9689    0.7355      3214
              normal     0.9817    0.9066    0.9426      2783
          depression     0.9872    0.8073    0.8882      3046
            suicidal     0.7595    0.1630    0.2684      2209

            accuracy                         0.7805     20698
           macro avg     0.8258    0.7568    0.7530     20698
        weighted avg     0.8226    0.7805    0.7695     20698
```

Figure 10: Screenshot of Evaluation Result of Random Forest

8.3 Model Evaluation Result of Naive Bayes

6.3 Naive Bayes Evaluation

```python
# Make predictions
y_pred_nb = nb_model.predict(X_test_tfidf)
y_pred_nb_proba = nb_model.predict_proba(X_test_tfidf)

# Calculate metrics
nb_accuracy = accuracy_score(y_test, y_pred_nb)
nb_precision = precision_score(y_test, y_pred_nb, average='weighted')
nb_recall = recall_score(y_test, y_pred_nb, average='weighted')
nb_f1 = f1_score(y_test, y_pred_nb, average='weighted')

print("NAIVE BAYES - PERFORMANCE METRICS")
print(f"\nAccuracy:  {nb_accuracy:.4f} ({nb_accuracy*100:.2f}%)")
print(f"Precision: {nb_precision:.4f}")
print(f"Recall:    {nb_recall:.4f}")
print(f"F1-Score:  {nb_f1:.4f}")

# Detailed classification report
print("DETAILED CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred_nb, target_names=class_names, digits=4))
```

```
NAIVE BAYES - PERFORMANCE METRICS

Accuracy:  0.7806 (78.06%)
Precision: 0.8178
Recall:    0.7806
F1-Score:  0.7896
DETAILED CLASSIFICATION REPORT
                     precision    recall  f1-score   support

            anxiety     0.8674    0.8110    0.8382      3524
            bipolar     0.9797    0.8253    0.8959      2742
             stress     0.4904    0.8028    0.6089      3180
personality disorder     0.7429    0.7184    0.7305      3214
             normal     0.9847    0.9260    0.9544      2783
         depression     0.9799    0.8145    0.8896      3046
           suicidal     0.6840    0.5048    0.5809      2209

           accuracy                         0.7806     20698
          macro avg     0.8184    0.7718    0.7855     20698
       weighted avg     0.8178    0.7806    0.7896     20698
```

Figure 11: Screenshot of Evaluation of Naive Bayes Result

8.4 Word Cloud of Text

2.3 Word Clouds

```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))

# Combine all text across statuses
all_text = " ".join(df['text'].tolist())

# Generate one word cloud
wc = WordCloud(width=2000, height=1000, background_color='white',
               stopwords=stop_words, max_words=350,
               colormap='viridis').generate(all_text)
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Figure 12: Screenshot Word Cloud

8.5 Class Distribution in Dataset

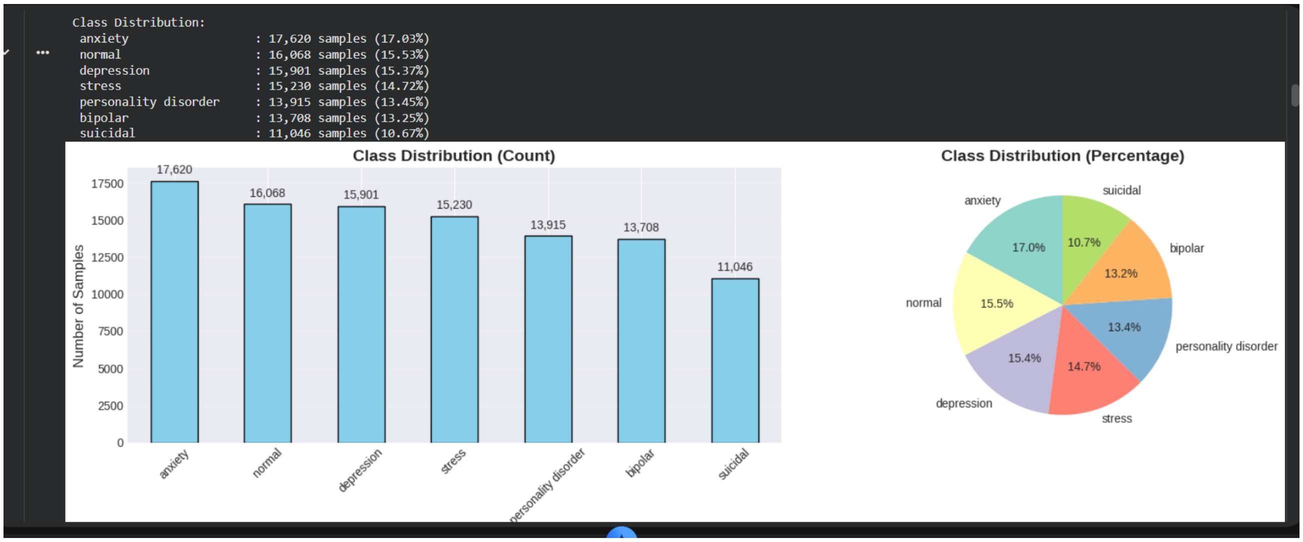


Figure 13 Screenshot of Class Distribution in Dataset

8.6 Model Performance Comparison

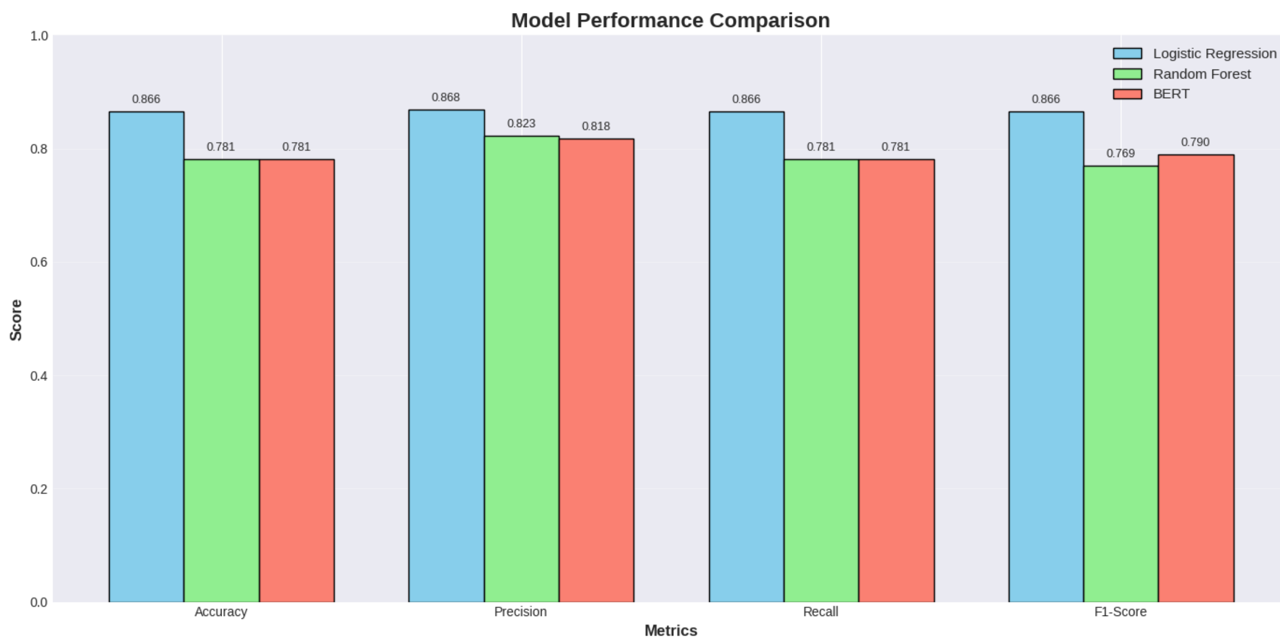


Figure 14: Screenshot of Comparison of Model Performance

8.7 Confusion Matrices of Logistic Regression

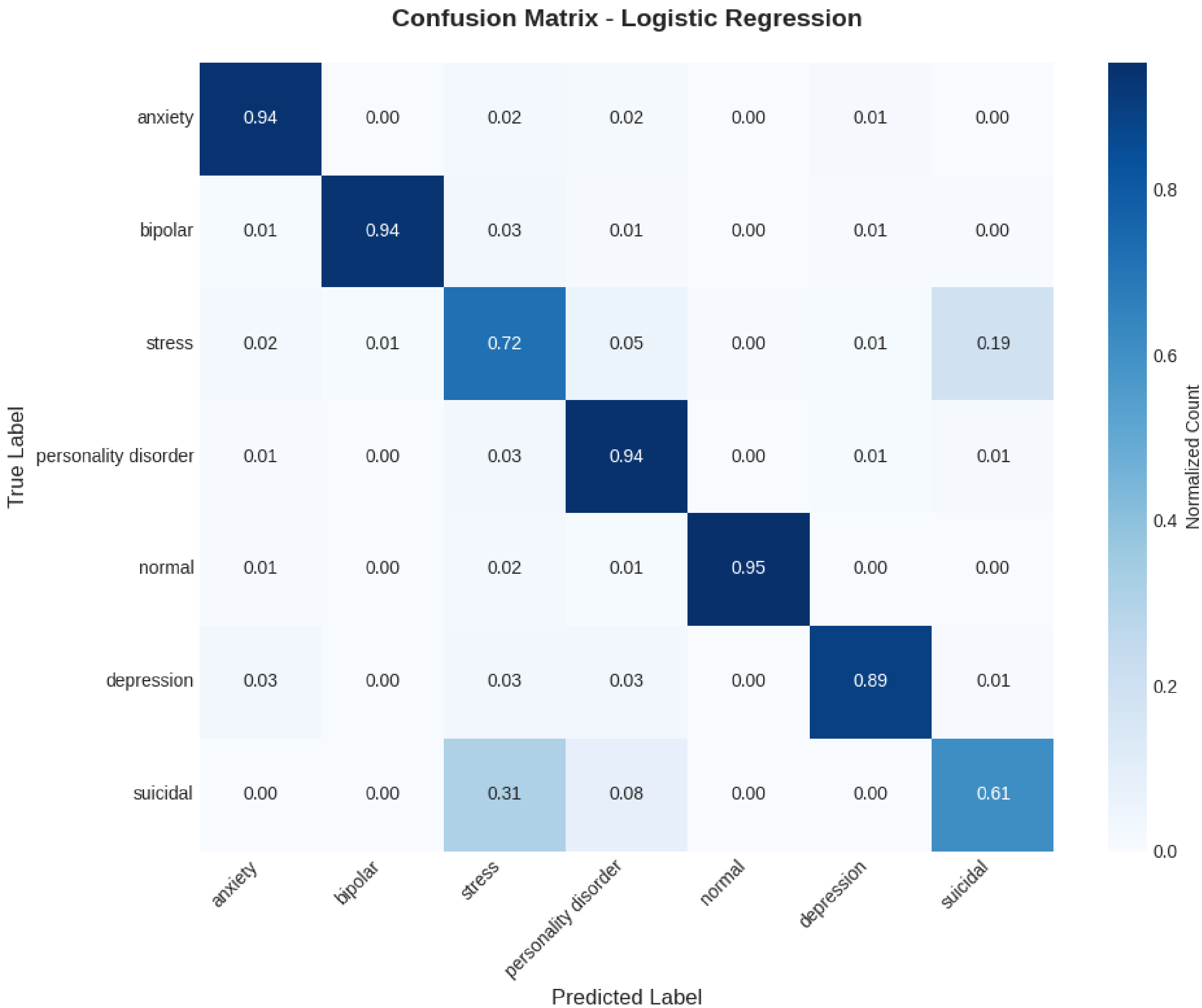


Figure 15: Screenshot of Confusion Matrices of Logistic Regression

8.8 Confusion Matrices of Random Forest

Figure 16: Screenshot of Confusion Matrix of Random Forest

8.9 Confusion Matrix of Naive Bayes

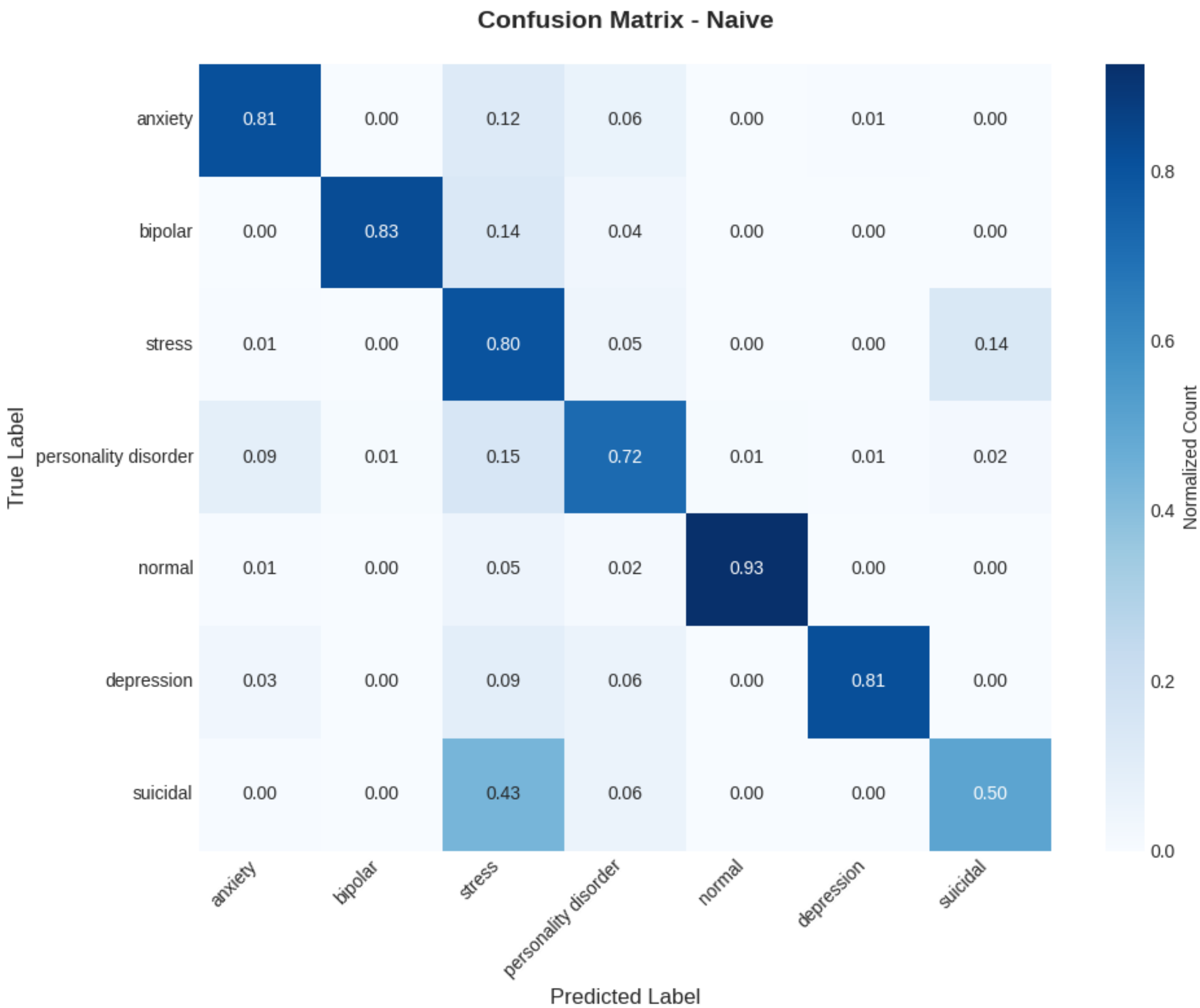


Figure 17: Screenshot of Confusion Matrix of Naive Bayes

8.10 ROC Curve of Logistic Regression

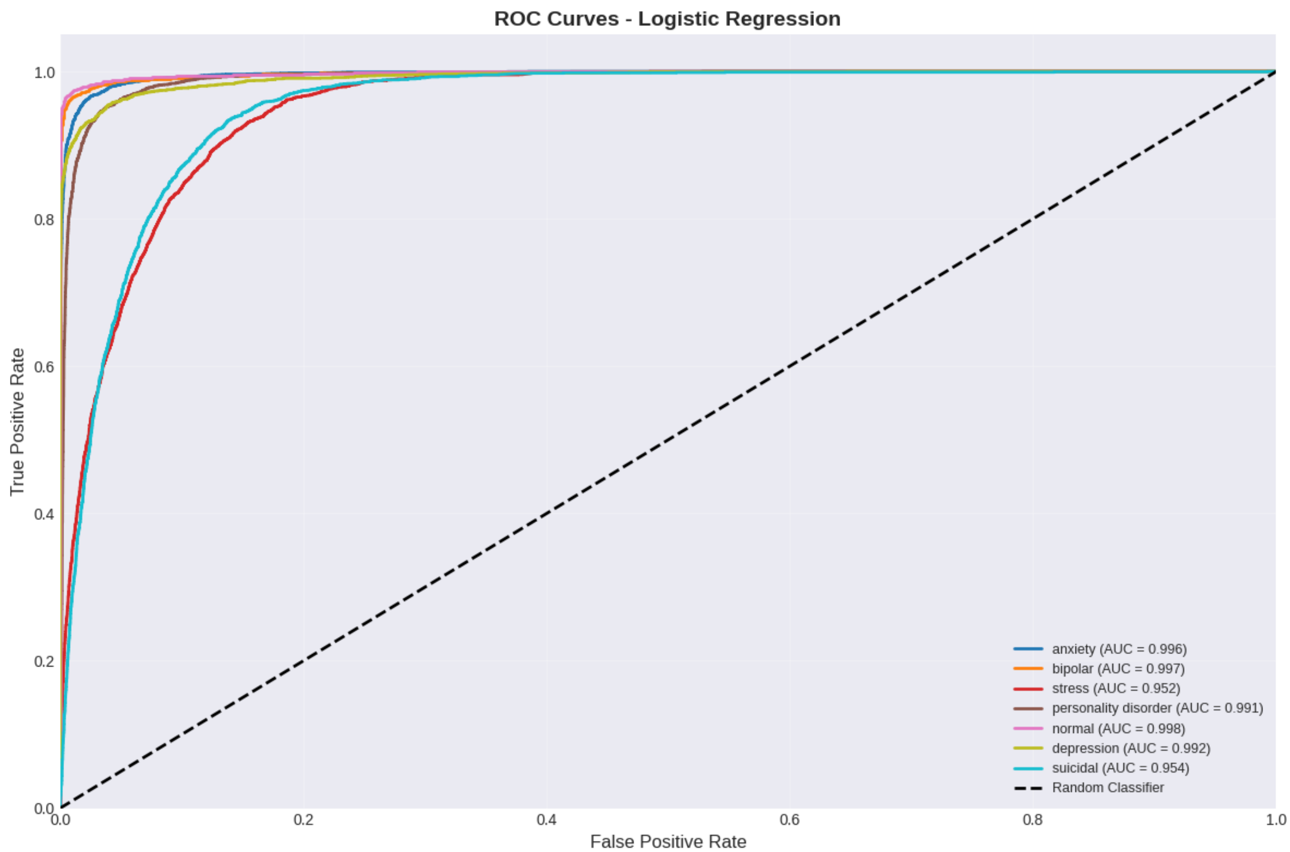


Figure 18: Screenshot of ROC Curve of Logistic Regression

8.11 ROC Curve of Random Forest

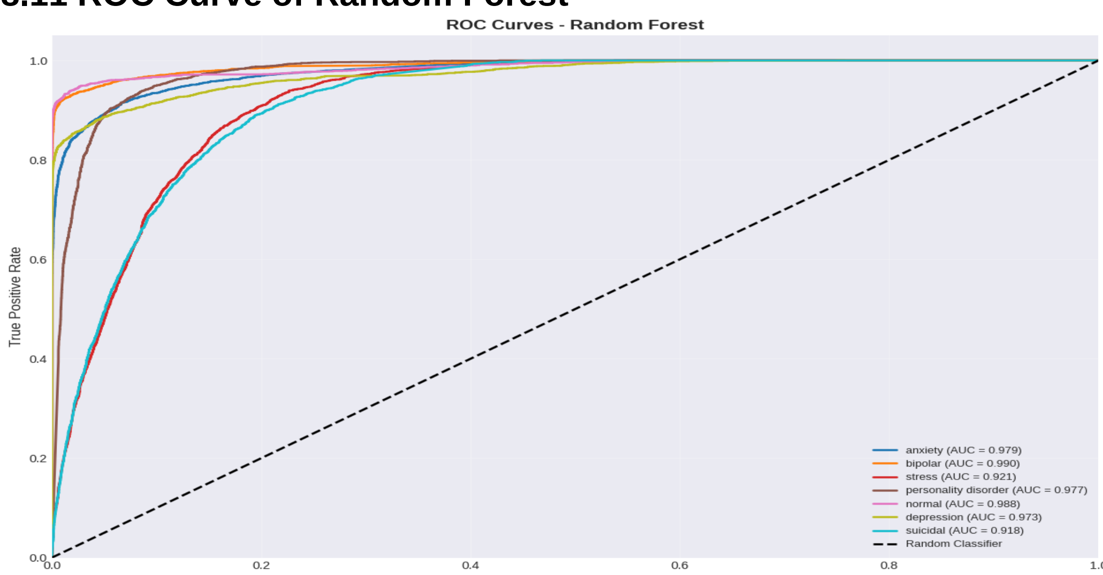


Figure 19: Screenshot od ROC Curve of Random Forest

8.12 ROC Curve of Naive Bayes

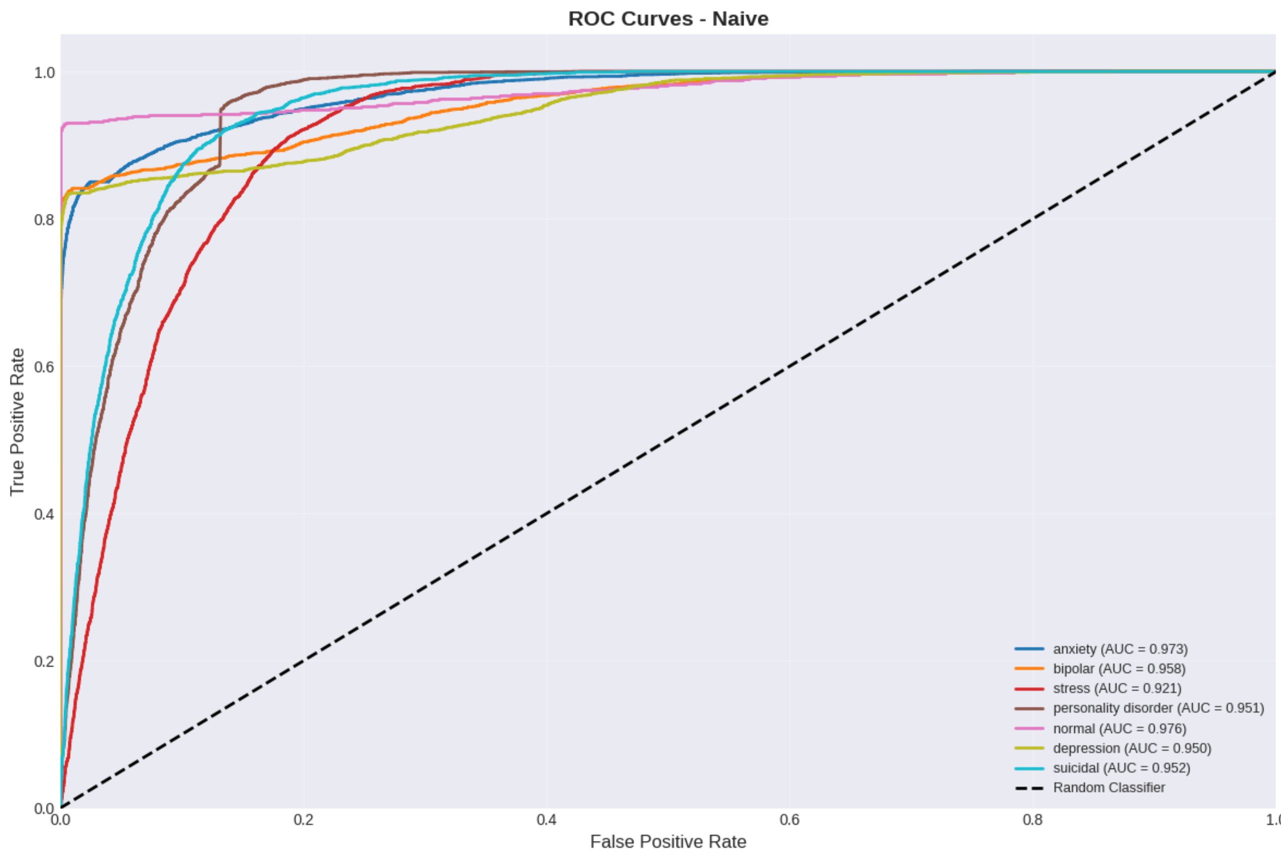


Figure 20: Screenshot od ROC Curve of Naive Bayes

8.13 Comparative Analysis of Model Performance

8.13.1 Difference in Performance

The three models implemented had demonstrated varying levels of classification performance across the seven class of mental health categories. Logistic Regression have achieved height AUC score, with values exceeding 0.99 while random forest closely followed logistic regression performance with AUC score 0.97. The naive bayes however perform little low in comparison with other models with the score ranging in between 0.92 to 0.95.

8.13.2 Possible Reasons for Performance Differences

The high performance of the logistic regression could be because of its ability to model linear decision boundaries effectively in high dimensional TF-IDF. Similarly, Random Forest’s high performance is very justifiable as it train the data in numerous decision tree and ensemble the output data. While Naive bayes assumes features independence which is rarely true in NLP this could limit its ability to predict the correct data with high accuracy.

8.13.3 Observations on Overfitting and Underfitting

Logistic Regression and Naive Bayes doesn’t show any sign of overfitting as their performance remained stable across the training and testing data sets. Random Forest might have a overfitting as deep trees and large features increases the training time and slightly reduces generalization on minority classes. Therefore, all models are benefited from consistent preprocessing and feature extraction. The result of all models are appreciable.

8.14 Best Model in Action (Predicting Sample Text)

```
Selected best model: Logistic Regression (score = 0.9830)

SAMPLE 1

INPUT TEXT:
I feel overwhelmed with constant worry and fear about everything in my life.

Model used: Logistic Regression
Predicted Class: anxiety
Confidence: 87.01%

SAMPLE 2

INPUT TEXT:
I'm experiencing extreme mood swings from very high energy to deep sadness.

Model used: Logistic Regression
Predicted Class: bipolar
Confidence: 80.37%

SAMPLE 3

INPUT TEXT:
I want to die.

Model used: Logistic Regression
Predicted Class: suicidal
Confidence: 89.75%
```

Figure 21: Screenshot of Model in Action

9. CONCLUSION

The Mental Health Condition Classification project proves that it is possible and efficient to use machine learning and natural language processing methods in the analysis of psychological conditions that are described in text. The creation of an end-to-end pipeline that connects preprocessing, feature extraction, and classification models allowed us to convert raw and noisy textual data into structured information on the mental health conditions.

Classical algorithms like Logistic Regression, Random forest and Naive Bayes were also very strong baselines when trained on TF-IDF features and this underscores the interpretability and effectiveness of linear methods in high-dimensional text spaces. The comparative analysis confirmed that the old approaches are still useful in terms of transparency and speed, but new transformer architecture is the standard of accuracy and generalization. Therefore, the proposed system developed a mental health text classification system that automatically classify the text to various categories.

Bibliography

Aggarwal, C. (2018). *Machine Learning for Text*. Springer.

Bentley, F., et al. (2020). Ethical considerations in social media mental health research. *Proceedings of the CHI Conference on Human Factors in Computing Systems*.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

Calvo, R. A., Milne, D. N., Hussain, M. S., & Christensen, H. (2017). Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, 23(5), 649–685.

Chancellor, S., & De Choudhury, M. (2020). Methods in predictive techniques for mental health status on social media: a critical review. *npj Digital Medicine*, 3(1), 43.

Coppersmith, G., Dredze, M., & Harman, C. (2015). Quantifying mental health signals in Twitter. *ACL Workshop on Computational Linguistics and Clinical Psychology*.

De Choudhury, M., Counts, S., & Horvitz, E. (2013). Predicting depression via social media. *ICWSM*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

Hugging Face (2025). *Mental Health Condition Classification Dataset*. Available at: https://huggingface.co/datasets/sai1908/Mental_Health_Condition_Classification.

Inkster, B., et al. (2018). Digital health interventions for mental health: evidence review. *JMIR Mental Health*.

Joachims, T. (1998). Text categorization with Support Vector Machines: learning with many relevant features. *European Conference on Machine Learning*.

Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft).

Lazer, D., et al. (2018). The science of fake news. *Science*, 359(6380), 1094–1096.

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Miner, A. S., et al. (2020). Smartphone-based conversational agents and mental health. *JAMA*.