UNIVERSITY OF MASSACHUSETTS DARTMOUTH

DEPARTMENT OF DATA SCIENCE

DSC 550: MASTER'S PROJECT

AI-Powered Mental Health Diagnostic Aid Through Language Analysis

Project Advisor

Dr. Amir Akhavan Masoumi

Presented By

Sourabh Deelip Pratapwar

Student Id: 02129773

Table of Content

ABSTRACT......1
ACKNOWLEDGMENTS......2
Chapter 1: INTRODUCTION......3
1.1 Motivation......3
1.2 Objectives......3
Chapter 2: SYSTEM ARCHITECTURE......4
2.1 Data Acquisition Module......4
2.2 Preprocessing Module......4
2.3 Feature Extraction Module......4
2.4 Machine Learning Module......4
2.5 Report Generation Module......4
2.6 User Interface Module......5
Chapter 3: DATA COLLECTION AND PREPROCESSING......6
3.1 Data Collection......6
3.2 Preprocessing Techniques......6
Chapter 4: FEATURE EXTRACTION......9
4.1 Sentiment Analysis......9
4.2 Keyword Extraction......9
4.3 Tone & Emotion Detection......9
4.4 Integration with the Classification Pipeline......10
Chapter 5: METHODOLOGY......11
5.1 Logistic Regression......11
5.2 Support Vector Machine (SVM)......11
5.3 LSTM (Long Short-Term Memory)......12
5.4 BERT (Bidirectional Encoder Representations from Transformers)......13
Chapter 6: PERFORMANCE METRICS AND EVALUATION......15
6.1 Performance Metrics for Logistic Regression......15
6.2 Performance Metrics for SVM......16
6.3 Performance Metrics for LSTM......17
6.4 Performance Metrics for BERT Implementation......18
6.5 Model Selection......20
Chapter 7: REPORT GENERATION AND UI DESIGN......21

Chapter 8: RESULTS **22**
Chapter 9: CONCLUSION **24**
Chapter 10: FUTURE WORK **25**
REFERENCES **26**
APPENDIX **27**
APPENDIX A: Supervised Machine Learning Models (Supervised_ML_Models.ipynb) **27**
APPENDIX B: BERT Model (BERT_Model.ipynb) **29**
APPENDIX C: Deployment App (final_app.py) **31**

List Of Figures

Figure 1: System Architecture 5
Figure 2: Preprocessing Concept Diagram..8
Figure 3: Performance comparison of models for multi-label mental health classification 20
Figure 4: System Workflow for Clinical Dashboard Integration 21

List Of Tables

Table 1: Classification Report of Logistic Regression 15
Table 2: Confusion Matrix of Logistic Regression....................15
Table 3: Classification Report of SVM....................16
Table 4: Confusion Matrix of SVM....................16
Table 5: Classification Report of LSTM....................17
Table 6: Confusion Matrix of LSTM....................17
Table 7:Training and Validation Trends for BERT Model....................18
Table 8: Classification Report of BERT Model 19
Table 9:Confusion Matrix of BERT Model 19

ABSTRACT

Mental health problems such as stress and anxiety and depression exist widely but their diagnosis remains difficult because symptoms appear differently in each person. Our research develops an AI tool which examines written text to detect faint indicators of these conditions. The system uses advanced natural language processing methods to detect emotional undertones and sentiment and pivotal keywords which reveal significant patterns that might indicate underlying issues. The system achieves its final form through extensive training on meticulously curated and labeled datasets which produces easy-to-understand reports that help clinicians identify critical areas for closer examination and support during treatment.

ACKNOWLEDGMENTS

I'm truly grateful to my advisor for his guidance, mentorship, and encouragement throughout this project. His expertise and thoughtful feedback not only shaped the way I approached the research but also ensured it stayed relevant both technically and clinically.

The open-source NLP community provided essential tools and frameworks which enabled this work to become possible. The development and fine-tuning of models became faster using Hugging Face Transformers together with Scikit-learn and NLTK libraries.

I would like to thank the developers of public mental health datasets including the Reddit Mental Health Data on Kaggle because they supplied essential resources for training and testing the system. The work benefits from structured diagnostic knowledge of mental health researchers and practitioners who continue to inspire innovations in mental health care.

Chapter 1: INTRODUCTION

The diagnosis of mental health conditions starting with stress and anxiety and depression has proven difficult because it heavily depends on personal judgments and reported symptoms of patients. Clinicians who possess training in psychological distress identification face challenges because of how patients communicate and how they interpret symptoms and limited time available for assessment.

Written language contains substantial information about how people feel mentally. The language choices people make together with their writing tone and repeating topics indicate minor behavioral signs which can connect to mental health disorders. Modern natural language processing techniques enable researchers to analyze psychological indicators in a methodical fashion through data analysis. This research project develops an AI diagnostic system which evaluates written text to detect mental health indicators so healthcare professionals can base their decisions on more objective data instead of relying solely on personal judgments.

1.1 Motivation

The leading causes of disability across the world include mental health disorders but numerous people fail to receive diagnoses until their conditions reach advanced stages. People face delays in treatment because they encounter several obstacles which include restricted mental health access and high consultation costs and social stigma and unreliable evaluation techniques.

The implementation of AI technology for written language analysis creates an opportunity to establish uniform preliminary evaluations while detecting early warning indicators through objective methods. The tool functions as an initial diagnostic component which performs written input screening to identify potential issues while generating organized assessment data for clinical professionals. The implemented approach facilitates both time efficiency and enables prompt interventions which play an essential role in enhancing treatment results.

1.2 Objectives

- The system should help medical staff identify mental health indicators through written language pattern analysis at early stages.
- The system should produce readable reports which present analytical results alongside specific therapeutic investigation recommendations.
- The diagnostic process will achieve greater efficiency along with standardized results because the system reduces the first stage assessment variations.

Chapter 2: SYSTEM ARCHITECTURE

The proposed diagnostic aid functions as a modular system which operates by converting basic written input into functional clinical insights through specific components. The modular structure of this system provides flexibility and scalability which makes maintenance and updates and adaptation to new requirements easier in the long run.

The system consists of six core modules:

2.1 Data Acquisition Module

The system gathers written content which patients submit for processing. Patients can enter responses through structured questionnaires as well as open-ended journal-style entries and short self-reflection prompts. The system aims to collect material which represents the natural expression of thoughts and emotions and linguistic patterns. The collected data remains protected in privacy-compliant storage systems which ensure the security of sensitive patient information.

2.2 Preprocessing Module

Text analysis requires preprocessing as the first step before any analysis can proceed. The system performs text cleaning operations followed by lemmatization to convert words into their base forms and tokenization to split the text into manageable units for machine learning algorithms. The process of noise elimination and standardized formatting enhances the accuracy of following analysis stages.

2.3 Feature Extraction Module

This system uses multiple natural language processing (NLP) techniques to reveal significant patterns within the data. The analysis of sentiment helps determine if the overall text expresses positive or negative or neutral emotions. The keyword extraction function identifies particular words which have associations with emotional states as well as mental health markers. The system moves beyond basic emotion detection to identify tones such as sadness or anxiety or hopelessness that require clinical evaluation.

2.4 Machine Learning Module

This is the analytical engine of the system. The system applies multi-label classification to detect multiple conditions including depression anxiety and stress in a single text input. The models undergo training with labeled datasets to detect refined language patterns that correspond to each condition.

2.5 Report Generation Module

This module consolidates all analysis results into structured reports that users can easily understand. The report generates emotional tone identification alongside keyword frequency analysis and a summary of probable health conditions alongside therapeutic area recommendations. The outputs are designed to provide interpretable results which help clinicians swiftly understand and apply the findings.

2.6 User Interface Module

The system's last section displays insights through a clinician-friendly dashboard which organizes information in an easy-to-understand format. Healthcare providers can access patient reports and specific indicators through a simple few-step process after reviewing summary information. The interface has been designed to blend seamlessly with established workflows which allows for minimal disruption alongside enhanced usability.

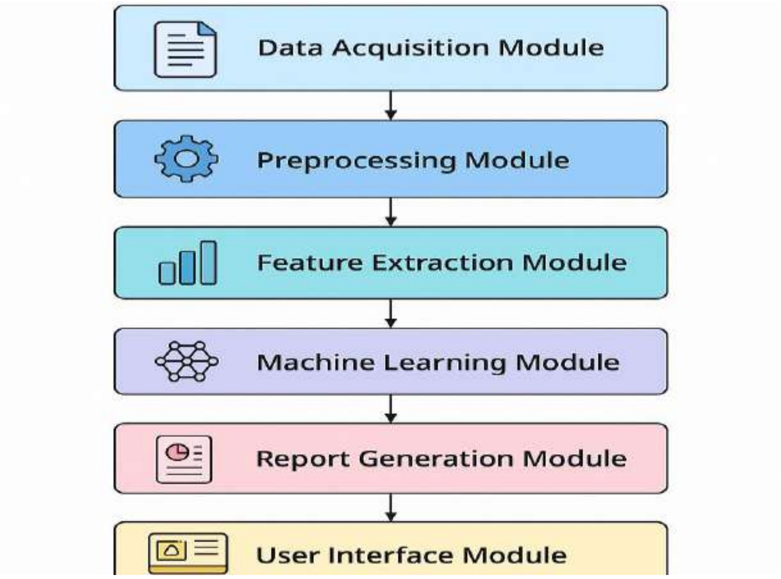


Figure 1: System Architecture

Chapter 3: DATA COLLECTION AND PREPROCESSING

3.1 Data Collection

The project utilized the Reddit Mental Health Data collection on Kaggle which aggregates content from mental health-focused subreddits for data analysis. These virtual communities function as open platforms which let users share their life experiences along with their emotional struggles and mental health challenges to generate authentic real-world language data.

The text entries in the dataset receive their mental health condition labels through the following mapping system:

0 – Stress
1 – Depression
2 – Bipolar Disorder
3 – Personality Disorder
4 – Anxiety
The dataset's label structure aligns perfectly with multi-label classification since a single post can demonstrate symptoms of multiple conditions simultaneously.

The dataset received additional examples which were reviewed by mental health professionals for proper annotation. The process resulted in better edge case representation for indirect or metaphorical expressions of distress which enhanced the model's ability to interpret complex language.

3.2 Preprocessing Techniques

Machine learning required preprocessing techniques to transform unstructured Reddit text into an organized format that could be used for analysis. The objective of this process was to maintain vital text components while eliminating disruptive elements that might interfere with model learning. The following section describes the preprocessing techniques with their application steps in the project pipeline.

3.2.1 Text Normalization

The text preprocessing operation converted all input text to lowercase to treat "Depression" and "depression" as identical words. The removal process eliminated punctuation marks together with numbers and non-alphabetic symbols. The preprocessing steps utilized Python regular expressions (`re.sub`) and standard string manipulation methods for implementation. Standardization of text data decreased vocabulary space which led to better model performance and minimized feature set sparsity.

3.2.2 Stop Word Removal

The NLTK library's stop word list received modifications to keep words related to mental health importance including "not" because it changes sentiment. The model processed contextually important words by excluding common terms that provided little value.

3.2.3 Lemmatization

Lemmatization transformed words into their dictionary roots to allow the model to treat different word forms as equivalent. The words "feeling" and "feels" were both converted into "feel" through this process. WordNetLemmatizer from NLTK performed this operation. The model required lemmatization instead of stemming because it maintained more precise base forms which proved essential for mental health language understanding.

3.2.4 Tokenization

Before performing feature, extraction operations text was converted into separate word tokens. NLTK's `word_tokenize` function performed tokenization of text to create separate tokens for subsequent analysis. Tokenization enabled the application of vectorization techniques such as TF-IDF which were used in both the `Supervised_ML_Models.ipynb` and `Multilabel_classification_model.ipynb` notebooks to transform words into numerical features.

3.2.5 Handling Class Imbalance

The preprocessing process included methods to manage the unequal distribution of labels which appeared in the dataset. The Multilabel_classification_model.ipynb notebook used Synthetic Minority Oversampling (SMOTE) along with other oversampling methods to achieve label balance between bipolar disorder and Depression as well as other mental health conditions.

3.2.6 Vectorization

After applying cleaning operations and tokenization to the data the processed text underwent TF-IDF Vectorization to generate numerical features. Through this method words receive weights which represent their importance in individual documents relative to all documents in the dataset thus enabling better detection of unique and condition-specific language patterns. The project implemented a max feature limit to prevent overfitting while including n-grams (bigrams and trigrams) to detect short diagnostic phrases like "feeling hopeless" or "can't cope" which hold strong diagnostic value.

The preprocessing techniques were applied precisely to create inputs which were both linguistically clean and semantically rich for the models so the algorithms could learn patterns related to stress, depression, bipolar disorder, personality disorder and anxiety.

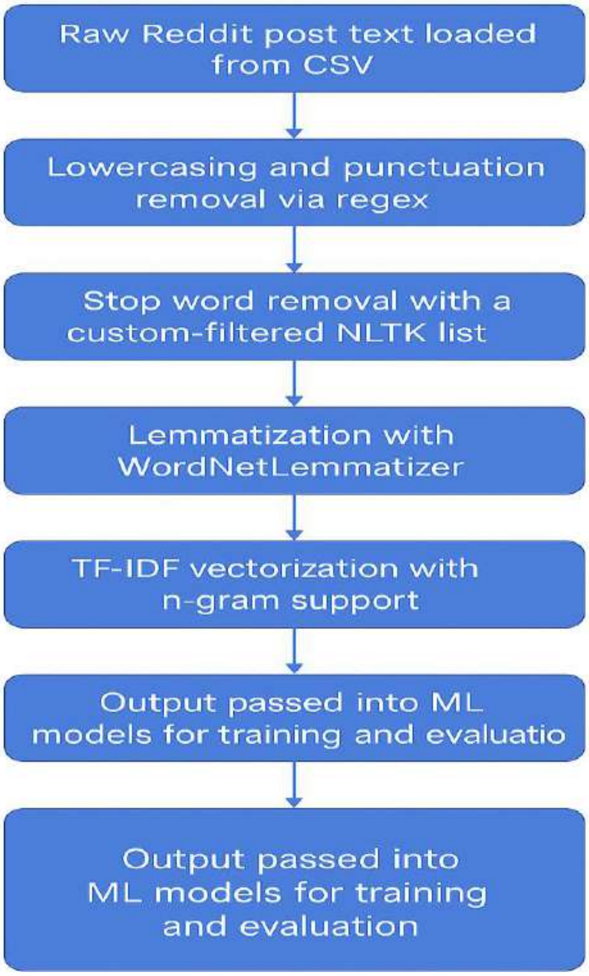

Raw Reddit post text loaded from CSV
Lowercasing and punctuation removal via regex
Stop word removal with a custom-filtered NLTK list
Lemmatization with WordNetLemmatizer
TF-IDF vectorization with n-gram support
Output passed into ML models for training and evaluatio
Output passed into ML models for training and evaluation

Figure 2: Preprocessing Concept Diagram

Chapter 4: FEATURE EXTRACTION

The feature extraction stage functions as a vital operational step since it turns text data from preprocessing into numerical representations that machine learning systems can understand. Three main approaches including sentiment analysis and keyword extraction and tone & emotion detection with transformer-based embeddings were implemented in this project.

4.1 Sentiment Analysis

The method of sentiment analysis determines the total emotional direction of written text which exists between positive and negative and neutral states. The emotional strength of text content in mental health settings can be determined by sentiment polarity analysis because it shows the degree of emotional distress.

This project uses transformer embeddings for sentiment analysis instead of traditional analyzers like VADER because transformer embeddings provide superior contextual understanding. Sentiment cues are automatically included in BERT embeddings during classification so a separate sentiment classifier was unnecessary although sentiment signals from the embeddings and model logits can still be extracted.

4.2 Keyword Extraction

The process of keyword extraction helps to identify specific terms which strongly relate to mental health states. The words "hopeless", "panic" and "overwhelmed" serve as examples.

During the preprocessing step TF-IDF vectorization with n-gram support (unigrams, bigrams, and trigrams) was utilized. This approach evaluates word and short phrase importance through their document frequency relative to their total corpus frequency. The TF-IDF score indicates the degree of importance of words or phrases for mental health indicator detection.

The system detects single words as well as short multi-word expressions which prove essential for diagnosing conditions (e.g., "can't sleep," "feel worthless").

4.3 Tone & Emotion Detection

This project implements emotion detection through BERT-based embeddings which represent its most complex feature extraction method. Hugging Face's Transformers library offers `bert-base-uncased` as the foundation for this approach.

The process involved:

i. The BERT tokenizer performed text tokenization.
ii. Tokenization leads to the generation of embedding representations that retain semantic and emotional aspects of the text.
iii. The model used fine-tuned BertForSequenceClassification architecture to predict multiple mental health conditions in a multi-label setup after receiving input embeddings.

Through this technique the model detects minute language indicators including tone expressions and metaphorical expressions and self-description patterns which basic word counting methods fail to recognize.

4.4 Integration with the Classification Pipeline

After extraction the machine learning module receives features including sentiment scores and top keywords and deep contextual embeddings. The BERT model performs embedding and classification steps in a unified end-to-end system which eliminates the need for manual feature development.

Chapter 5: METHODOLOGY

5.1 Logistic Regression

Overview of Logistic Regression

Logistic Regression functions as a supervised machine learning technique which performs classification operations. The logistic (sigmoid) function transforms linear combinations of input features into probabilities between 0 and 1 to model class membership. The model fits independent logistic models for each label to enable simultaneous prediction of multiple conditions. The model demonstrates two main advantages: it provides interpretable results through coefficients that show feature importance, and it operates efficiently in large-dimensional spaces like TF-IDF text vectors.

Use of Logistic Regression in the Project

The project uses Logistic Regression as its baseline classifier to identify mental health conditions including stress and depression alongside bipolar disorder and personality disorders and anxiety. The model works with TF-IDF-transformed Reddit post text data through TF-IDF encoding which represents word significance against the corpus while maintaining high-dimensional sparsity. The text feature vectors receive probability mappings from Logistic Regression through its learned weights which reveal interpretable linguistic markers such as "hopeless" and "panic" as strong indicators of depression and anxiety.

Model Training Details

The training process starts by dividing the cleaned preprocessed dataset into an 80/20 train-test ratio. The model uses a high iteration limit of max_iter=1000 to achieve convergence when working with the extensive sparse TF-IDF space. The multi-label setup trains each label independently through one-vs-rest optimization to achieve optimal classification results for each condition. The model receives TF-IDF vectors from the training data before it evaluates its performance on the test data through accuracy measurements and classification report (precision, recall, F1-score). The straightforward training process of this approach enables direct comparison with complex models including BERT and LSTM.

5.2 Support Vector Machine (SVM)

Overview of Support Vector Machine (SVM)

The Support Vector Machine (SVM) operates as a margin-based classifier through supervised learning algorithms to discover the optimal separating hyperplane between classes in feature space. The model selects a hyperplane that creates the largest margin between data points from different classes (support vectors) to enhance generalization performance. Linear SVMs prove highly effective for text classification because they handle sparse high-dimensional inputs (such as TF-IDF vectors) efficiently while achieving high accuracy without requiring complex kernel computations.

Use of SVM in the Project

The project implements SVM as its baseline traditional machine learning model to detect stress and depression alongside bipolar disorder and personality disorders and anxiety. The multi-label classification framework uses scikit-learn's LinearSVC to train separate classifiers for each label. The model processes Reddit text posts directly after TF-IDF transformation to detect the most significant words and phrases which distinguish positive from negative examples for each condition. The linear approach provides efficient computation with strong decision boundaries which makes it suitable for handling big textual datasets.

Model Training Details

The "Supervised_ML_Models.ipynb" implements SVM with default parameters ("LinearSVC()") before training with the TF-IDF feature matrix from the training set using "svm.fit(X_train_tfidf, y_train)". The dataset divides into 80/20 train-test ratio to maintain fair evaluation. The model generates predictions from the test data and evaluates them through accuracy measurements along with a classification report that shows precision and recall and F1-score for each mental health label. SVM provides interpretable non-probabilistic outputs through linear decision boundaries that function effectively in high-dimensional spaces although it lacks probabilistic outputs and deep contextual embeddings of transformer models like BERT.

5.3 LSTM (Long Short-Term Memory)

Overview of Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) network architecture operates as a particular recurrent neural network to address sequential data dependencies and solve the vanishing gradient problem which affects standard RNNs. The model uses three gates to manage information flow: input gates, forget gates and output gates which enable it to maintain important context across long sequences while eliminating noise. The specific design of LSTMs makes them ideal for natural language processing because sentences require word order and context to establish meaning.

Use of LSTM in the Project

The LSTM model functions as the deep learning reference system for detecting multiple mental health conditions including stress and depression and bipolar disorder and personality disorders and anxiety. The model analyzes Reddit mental health posts through temporal and contextual cues to detect co-occurring conditions beyond traditional TF-IDF-based models like Logistic Regression and SVM. The LSTM model uses sequential text analysis to detect delicate linguistic patterns and emotional changes which results in better mental health discourse representation.

Model Training Details

The LSTM pipeline starts with text preprocessing operations which include cleaning and tokenization and sequence padding to maintain consistent input length. The embedding layer transforms words into dense vector spaces before passing them to the LSTM layer. The network trains end-to-end through backpropagation through time (BPTT) with the Adam optimizer to optimize multi-label output. The final dense layer consists of five nodes which correspond to mental health conditions and uses sigmoid activation to generate independent probability scores. The model uses thresholding to convert output probabilities into binary predictions. The model design enables the detection of condition co-occurrences and generates probabilistic output predictions which enhance the transformer-based BERT model used in the project.

5.4 BERT (Bidirectional Encoder Representations from Transformers)

Overview of BERT

BERT functions as the deep learning backbone for mental health condition multi-label classification in this project. BERT differs from typical sequence models since its bidirectional transformer structure processes text in both left-to-right and right-to-left directions. BERT's dual perspective allows it to understand contextual details and detect subtle negations and sentiment changes which mental health discussions require for proper interpretation. A phrase containing "not feeling depressed anymore" requires analysis of context from both directions to achieve correct interpretation. The specific wording in clinical and self-reported mental health text requires BERT because it enables accurate interpretation.

Use of BERT in the Project

The implementation starts with the pre-trained "bert-base-uncased" model from Hugging Face Transformers which incorporates general English syntax and semantics understanding. The implementation fine-tunes this model using the custom Reddit Mental Health dataset to adapt it for the specific vocabulary and expression patterns found in discussions about stress and depression alongside bipolar disorder and personality disorders and anxiety.

A single sigmoid-activated neuron functions as each condition classification head in the system. The model predicts multiple conditions in each post by using five independent sigmoid-activated neurons because mental health conditions tend to appear together (e.g., anxiety with depression). The output predictions get converted to probability scores that become clinical-style reports by incorporating TF-IDF keywords and VADER sentiment analysis for both quantitative and qualitative interpretation.

Model Training Details

BertTokenizer processes raw text into WordPiece subword tokens which generate "input_ids", "attention_masks" and token type IDs that BERT transformer layers can use. The classification head receives its input from the "[CLS]" token pooling operation which generates contextual embeddings for each token across BERT transformer layers.

The training setup implements BCEWithLogitsLoss for multi-label classification along with the Adam optimizer and weight decay for stable learning. The optimization process requires adjustments to "max_seq_length" and "batch_size" and "learning_rate" to achieve the right balance between accuracy and efficiency. Training happens through repeated epochs while saving validation checkpoints until the model achieves acceptable performance on the held-out validation set. The threshold-sweeping strategy determines the probability-to-label conversions to maximize macro-F1 and recall which results in better performance across all conditions.

Deployment in Final Application

The application "final_app.py" integrates the trained model which receives its fine-tuning after completion. The application uses trained weights and tokenizer to analyze new user inputs in real-time operations. The diagnostic-style output combines the result of probability scoring with keyword extraction (TF-IDF) and sentiment analysis (VADER) which gets applied to text inputs after tokenization and model inference.

The contextual understanding and classification performance of BERT exceeds both Logistic Regression and SVM baseline methods. The project relies on this detection engine as its main engine to identify overlapping conditions and recognize subtle expressions which produces meaningful clinical results for healthcare providers.

Chapter 6: PERFORMANCE METRICS AND EVALUATION

6.1 Performance Metrics for Logistic Regression

- The Logistic Regression model reached an accuracy level of 76.02% when evaluating all five mental health conditions in the test set.
- The F1-score reached its highest value of 0.81 because bipolar disorder achieved equal precision and recall at 0.81.
- The model achieved its highest precision rate of 0.84 when detecting Stress which means it produces fewer incorrect positive predictions for this label.
- The F1-scores of all conditions remain close to each other with Depression and Personality Disorder at 0.72 and bipolar disorder at 0.81 which indicates no strong preference for any label.
- The model demonstrates equivalent performance through its macro-average and weighted-average F1-scores which both reach 0.76 despite the different class support sizes.
- The confusion matrix shows that the Logistic Regression model most accurately predicts Depression (197 correct) and Stress (162 correct), with relatively low misclassification rates across other categories. Some overlap exists, particularly between Depression and Personality Disorder, as well as Stress and Anxiety.

Class	Precision	Recall	F1-Score	Support
Stress	0.84	0.75	0.79	217
Depression	0.71	0.74	0.72	266
Bipolar Disorder	0.81	0.81	0.81	191
Personality Disorder	0.7	0.74	0.72	222
Anxiety	0.78	0.77	0.78	226
Accuracy			**0.76**	1122
Macro Avg	0.77	0.76	0.76	1122
Weighted Avg	0.76	0.76	0.76	1122

Table 1: Classification Report of Logistic Regression

Actual \ Predicted	Stress	Depression	Bipolar Disorder	Personality Disorder	Anxiety
Stress	162	14	4	16	21
Depression	12	197	14	29	14
Bipolar Disorder	1	20	154	10	6
Personality Disorder	6	37	7	165	7
Anxiety	13	11	10	17	175

Table 2: Confusion Matrix of Logistic Regression

6.2 Performance Metrics for SVM

- The SVM model reached 76.74% accuracy which surpassed the results of the Logistic Regression baseline.
- The F1-score of bipolar disorder reached 0.82 with precision and recall at 0.82 and 0.81 respectively which demonstrates balanced performance.
- The macro-average and weighted-average precision, recall and F1-score values are 0.77 which indicates that the model performs equally well on all classes.
- The model achieved excellent results for Anxiety detection through its 0.81 recall rate and 0.78 F1-score which demonstrates its strong performance in identifying anxiety-related content.
- The model demonstrates uniform performance across all classes because its F1-scores span between 0.73 and 0.82 without showing any preference for categories.
- The SVM model achieves good results in Depression (198 correct) and Anxiety (183 correct) classification but shows moderate confusion between Depression and Personality Disorder.
- The predictions for stress and bipolar disorder are strong (162 and 155 correct, respectively), though minor misclassifications occur across other categories. Cross-class confusion, particularly between Depression, Personality Disorder, and Anxiety, suggests overlapping linguistic features in the dataset.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Stress	0.82	0.75	0.78	217
Depression	0.73	0.74	0.74	266
Bipolar Disorder	0.82	0.81	0.82	191
Personality Disorder	0.73	0.73	0.73	222
Anxiety	0.76	0.81	0.78	226
Accuracy			**0.77**	1122
Macro Avg	0.77	0.77	0.77	1122
Weighted Avg	0.77	0.77	0.77	1122

Table 3: Classification Report of SVM

Actual / Predicted	**Stress**	**Depression**	**Bipolar Disorder**	**Personality Disorder**	**Anxiety**
Stress	**162**	16	7	14	18
Depression	13	**198**	8	30	17
Bipolar Disorder	1	21	**155**	5	9
Personality Disorder	10	27	7	**163**	15
Anxiety	12	8	11	12	**183**

Table 4: Confusion Matrix of SVM

6.3 Performance Metrics for LSTM

- The LSTM model reached only 23.71% accuracy which demonstrates its poor ability to generalize to test data.
- The model demonstrates an abnormal high recall rate of 0.97 for Depression which indicates it predicts this label in most cases thus creating an imbalance.
- The model demonstrates very poor performance in detecting Stress and Bipolar Disorder and Personality Disorder and Anxiety because recall values range between 0.00–0.02.
- The precision values remain low for all classes between 0.20–0.36 which indicates numerous incorrect positive predictions.
- The model demonstrates weak performance through both macro-average and weighted-average F1-scores which remain below 0.12.
- The LSTM model shows a severe bias toward predicting "Depression," misclassifying most other categories into this label, as reflected by the high counts in the second column (e.g., 211 Stress cases predicted as Depression).
- Minimal correct classifications occur across most classes, with notably poor performance in Personality Disorder and Anxiety, where true positives are almost non-existent.

Class	Precision	Recall	F1-Score	Support
Stress	0.29	0.01	0.02	217
Depression	0.24	0.97	0.38	266
Bipolar Disorder	0.36	0.02	0.04	191
Personality Disorder	0.2	0	0.01	222
Anxiety	0.22	0.01	0.02	226
Accuracy			**0.24**	1122
Macro Avg	0.26	0.2	0.09	1122
Weighted Avg	0.26	0.24	0.11	1122

Table 5: Classification Report of LSTM

Actual / Predicted	Stress	Depression	Bipolar Disorder	Personality Disorder	Anxiety
Stress	2	211	2	1	1
Depression	2	257	3	0	4
Bipolar Disorder	0	184	4	2	1
Personality Disorder	0	220	0	1	1
Anxiety	3	218	2	1	2

Table 6: Confusion Matrix of LSTM

6.4 Performance Metrics for BERT Implementation

Training and Validation Trends

BERT-based multilabel classification model training demonstrates predictable yet complex performance patterns during each epoch.

- The validation loss begins at 0.2918 before increasing to 0.3148 during the third training epoch. The model starts to overfit based on this minor increase which demonstrates the need to track early stopping criteria for maintaining generalization.
- The F1-macro score shows an upward trend during training because it rises from 0.7224 in epoch one to 0.7358 in epoch two before stabilizing at 0.7332 in epoch three.
- The model maintains its ability to achieve balanced precision and recall across all five mental health labels as the second epoch produces the best overall balance.
- The Macro precision value begins at 0.7499 and decreases slightly across epochs yet Macro recall increases from 0.6992 to 0.7427. The model improves its true positive detection capabilities at the expense of a minimal precision decrease which is acceptable for mental health classification because sensitivity remains crucial.
- The optimal classification threshold shows substantial changes during epochs because it decreases from 0.46 in the first epoch to 0.30 by the third epoch. The model adjusts its parameters dynamically to achieve maximum F1 performance which leads to better management of unbalanced labels and co-occurring conditions.

The BERT model demonstrates strong learning capabilities through its balanced precision and recall performance. The rising validation loss requires both epoch selection optimization and regularization methods to prevent overfitting and maintain generalizable results.

Epoch	**Step**	**Training Loss**	**Validation Loss**	**f1_macro**	**precision_macro**	**recall_macro**	**best_threshold**
0	500	None	0.2918	0.7224	0.7499	0.6992	0.46
1	1000	None	0.3024	0.7358	0.7329	0.7431	0.33
2	1326	None	0.3148	0.7332	0.7253	0.7427	0.3

Table 7: Training and Validation Trends for BERT Model

Performance Metrics:

The BERT-based implementation achieved high performance levels when detecting mental health conditions from Reddit text. The evaluation metrics demonstrate both the model's general reliability and its ability to detect various co-occurring conditions.

The Hamming Loss score of 0.11 indicates that the model predicted labels incorrectly in a minimal proportion. The model reached 88.9% label-wise accuracy which shows it performed well in classifying individual conditions. The model reached a Micro F1 score of 0.73 and a Macro F1 score of 0.73 which shows its ability to perform well on all classes. The Subset Accuracy (exact match) reached 69.9% which means that the model correctly identified seven out of ten posts even when multiple conditions co-occurred.

The class-level results show where the model performs well and where it faces difficulties. Stress received the highest identification rate from the model with an F1-score of 0.77 because of its strong precision and recall performance. The model demonstrated strong performance in bipolar disorder identification through its F1-score of 0.77 because it successfully recognized unique linguistic patterns in the data. The F1-scores for Anxiety and Personality Disorders reached 0.72 and 0.73 respectively. The model achieved better recall than precision in detecting depression cases with an F1-score of 0.69 and a recall of 0.73.

The confusion matrix reinforces these insights. The diagonal dominance in the matrix demonstrates that most instances received correct category assignments. The confusion matrix shows significant overlap between Depression and Anxiety as well as between Depression and Personality Disorders. Real-world clinical practice shows that these misclassifications occur because different mental health conditions frequently share similar linguistic and emotional content in written text.

The evaluation results demonstrate BERT's ability to detect subtle context and co-occurrence patterns which makes it an effective classifier for mental health condition detection beyond traditional machine learning baselines.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Stress	0.78	0.76	0.77	152
Depression	0.66	0.73	0.69	207
Bipolar Disorder	0.74	0.8	0.77	152
Personality Disorder	0.75	0.7	0.73	186
Anxiety	0.71	0.72	0.72	191
Accuracy			**0.73**	**888**
Macro Avg	0.73	0.74	0.73	888
Weighted Avg	0.72	0.74	0.73	888
Samples Avg	0.72	0.74	0.73	888

Table 8: Classification Report of BERT Model

Actual / Predicted	**Stress**	**Depression**	**Bipolar Disorder**	**Personality Disorder**	**Anxiety**
Stress	112	6	7	8	19
Depression	6	148	18	14	21
Bipolar Disorder	4	19	117	4	8
Personality Disorder	7	34	7	126	12
Anxiety	14	19	13	11	130

Table 9: Confusion Matrix of BERT Model

6.5 Model Selection

The evaluation results show BERT achieves superior performance than both traditional and deep learning models through its better accuracy and F1-score results. The performance of Logistic Regression and SVM maintained stable performance at moderate levels, but LSTM generated unstable results with decreased accuracy and F1 values.

BERT achieved the highest predictive accuracy of 0.889 and F1-score of 0.73 which proves its ability to make accurate predictions while maintaining strong resistance to class imbalances. The higher F1-score in this problem setting is crucial because it demonstrates BERT's ability to reduce both false positives and false negatives which is essential for applications with significant error implications.

The transformer-based attention mechanisms in BERT's architecture provide better contextual dependency understanding for text data than sequential models like LSTM. The model performs exceptionally well on complex tasks such as identifying multiple overlapping psychological conditions because it can identify delicate data details.

The graph shows BERT provides the best combination of reliability and generalization and robustness which made it the selected implementation model.

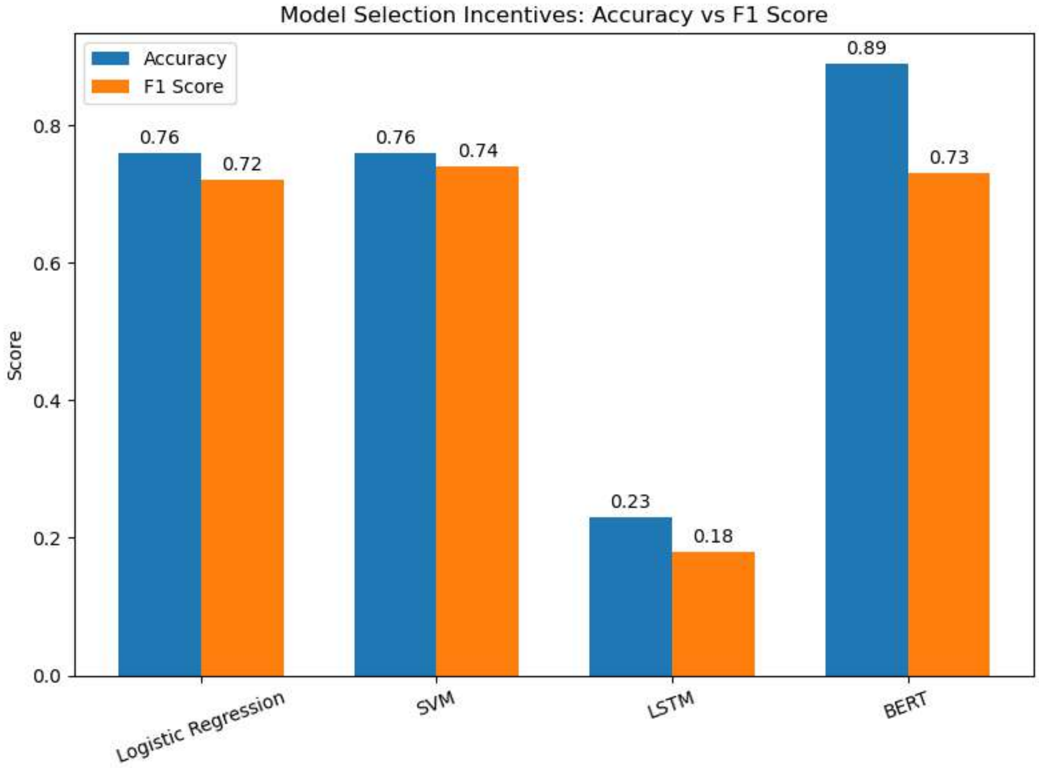


Figure 3: Performance comparison of models for multi-label mental health classification

Chapter 7: REPORT GENERATION AND UI DESIGN

The project requires transforming machine learning model predictions into clinical formats that healthcare providers can understand and apply in practice. The raw numbers together with technical outputs require transformation into understandable actionable insights. The development of report generation and user interface (UI) design represents a core component of this project. The system functions to connect sophisticated AI prediction outputs with the operational requirements of mental health professionals who practice in actual healthcare environments.

The report generation module integrates results from BERT, SVM and Logistic Regression models together with sentiment detection and keyword extraction and tone classification analyses. The system generates a summary for each patient text which includes predicted conditions along with confidence scores and essential keywords (e.g. "hopeless," "panic," "overwhelmed") that influenced the results. The analysis includes visual elements such as color-coded bars and probability charts which enable fast interpretation.

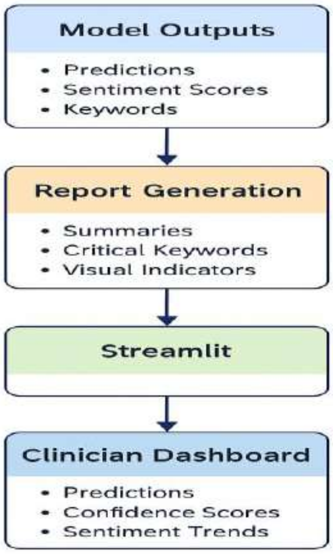
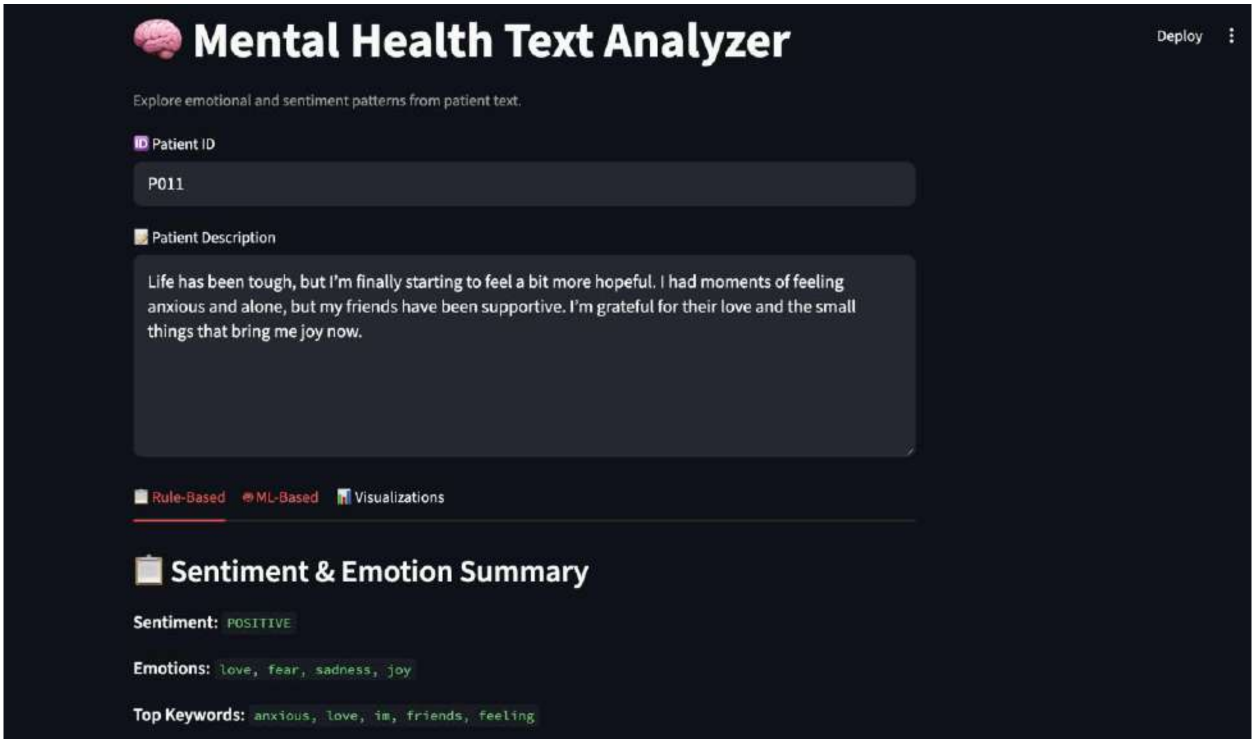


Figure 4: System Workflow for Clinical Dashboard Integration

Streamlit serves as the development platform for the UI which presents a simple and user-friendly interface. The system allows clinicians to upload text for analysis which produces organized results that display predictions and sentiment trends and keywords with detail exploration capabilities. The system provides transparent functionality with simple operation which transforms it into a dependable decision-support tool for better diagnostic speed and accuracy and confidence.

Chapter 8: RESULTS



🧠 Mental Health Text Analyzer

Explore emotional and sentiment patterns from patient text.

🆔 Patient ID

P011

📝 Patient Description

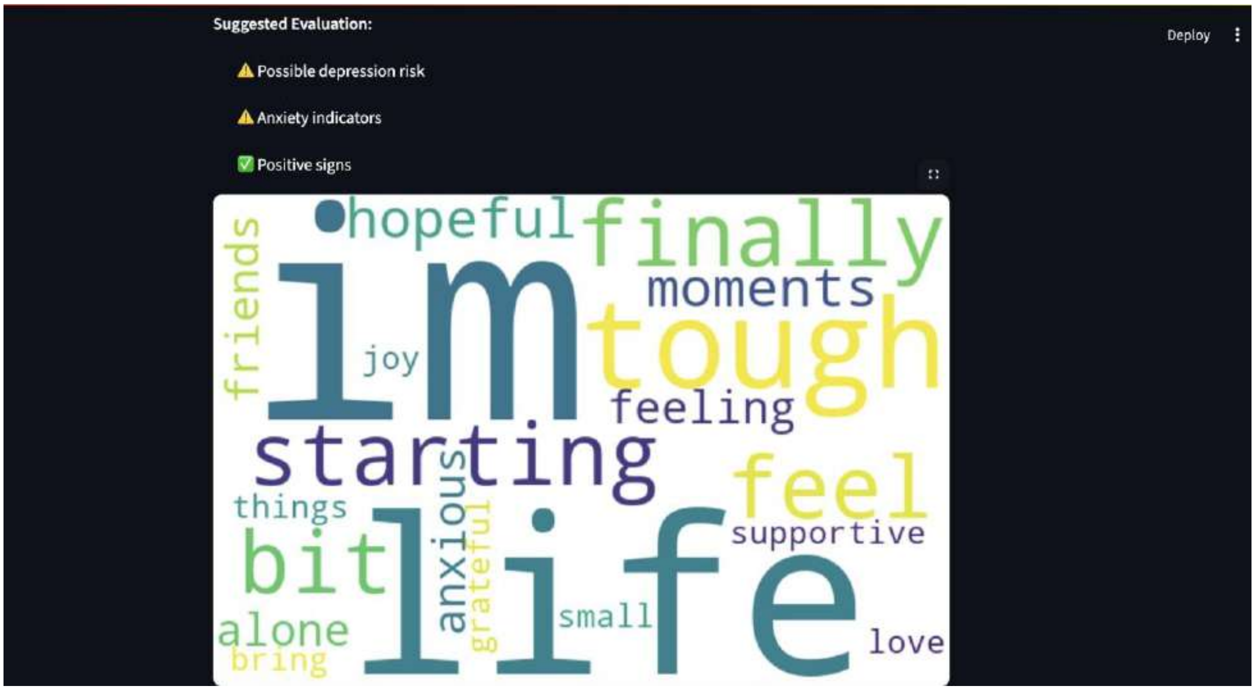
Life has been tough, but I'm finally starting to feel a bit more hopeful. I had moments of feeling anxious and alone, but my friends have been supportive. I'm grateful for their love and the small things that bring me joy now.

📋 Sentiment & Emotion Summary

Sentiment: `POSITIVE`

Emotions: `love, fear, sadness, joy`

Top Keywords: `anxious, love, im, friends, feeling`

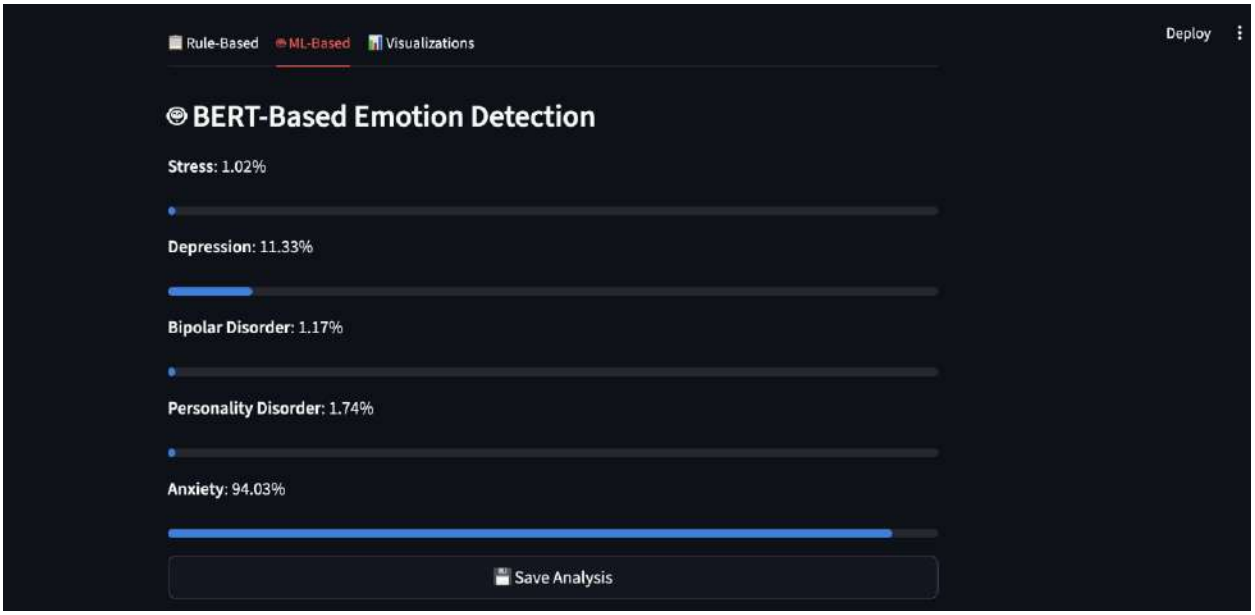


Suggested Evaluation:

⚠ Possible depression risk

⚠ Anxiety indicators

✅ Positive signs



BERT-Based Emotion Detection

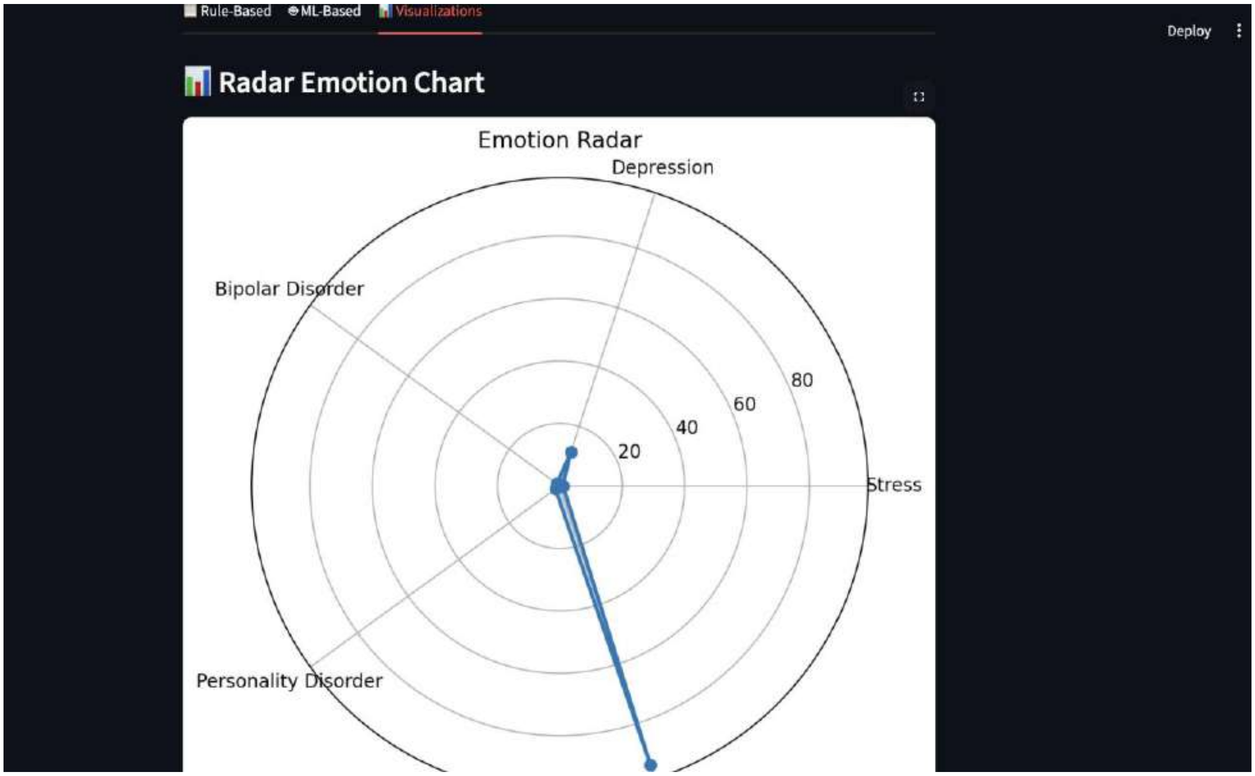
Stress: 1.02%

Depression: 11.33%

Bipolar Disorder: 1.17%

Personality Disorder: 1.74%

Anxiety: 94.03%



Radar Emotion Chart

Chapter 9: CONCLUSION

The research investigation confirmed that AI text analysis technology demonstrates potential to support mental health diagnosis. The system achieves this by using natural language processing techniques alongside multiple machine learning models to detect patterns in written text which indicate stress and depression and anxiety and bipolar disorder and personality disorders. The system reveals hidden patterns which medical staff would normally miss because it analyzes big text datasets at high speed with precision.

BERT and other classification models proved their ability to perform multi-label prediction with high accuracy which demonstrates AI systems can handle complex mental health conditions that often overlap. The system produces results that extend past basic prediction functions. The system presents results through an intuitive Streamlit-based interface and carefully designed report generation which produces findings that clinicians can understand and use to make decisions. The tool enhances patient well-being understanding through its presentation of predicted labels together with sentiment trends and confidence scores and key keywords.

The project demonstrates how AI functions as a helpful diagnostic tool which enhances early condition detection while improving clinical consistency and directing medical staff to critical assessment areas. Mental health assessment tools of this type have the potential to speed up the process while delivering better information and equal treatment for all patients.

Chapter 10: FUTURE WORK

1) Integration of Speech Emotion Recognition
   - The system should analyze spoken language elements including tone pitch and rhythm to detect emotional signals which text-based analysis cannot detect.
   - The system should use verbal and written communication analysis to improve its telehealth session assessment capabilities.

2) Expansion to Multilingual Datasets
   - The system should receive training from datasets which include various languages to provide service to different linguistic and cultural populations.
   - The system should enhance its inclusivity, and accessibility features to support global mental health support.

3) Deployment via Secure HIPAA-Compliant Platforms
   - The system needs to operate on secure platforms which follow regulations to protect patient privacy and maintain data protection standards.
   - The system needs to fulfill all legal and ethical standards for clinical environment integration.

4) Use of Explainable AI (XAI)
   - The system should provide transparency features to reveal which specific words and patterns and phrases affected the prediction results.
   - The system should establish clinician trust while enhancing patient communication and promoting responsible AI practices in healthcare.

REFERENCES

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

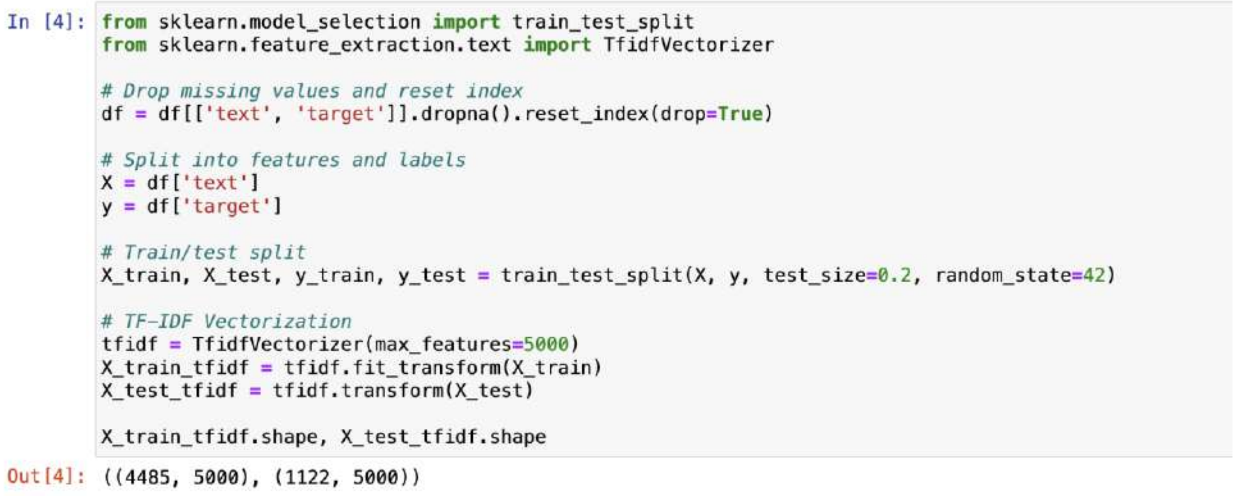
[8] N. Ghoshal, "Reddit mental health data," *Kaggle*, [Online]. Available: https://www.kaggle.com/datasets/neelghoshal/reddit-mental-health-data?resource=download.

[9] *EdrawMax Online Diagram Tool*, [Online]. Available: https://www.edrawmax.com/online/en/.

APPENDIX

APPENDIX A: Supervised Machine Learning Models (Supervised_ML_Models.ipynb)

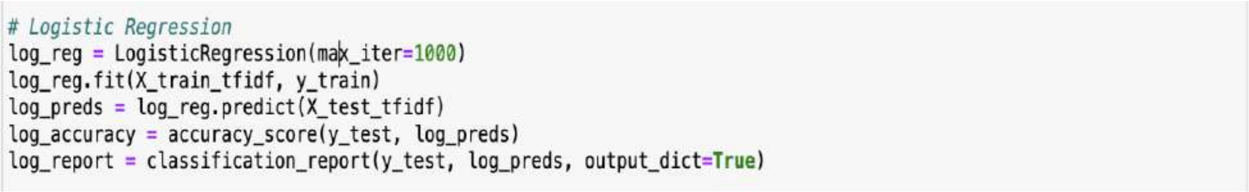
Data Preprocessing and Train-Test Split



```
In [4]: from sklearn.model_selection import train_test_split
        from sklearn.feature_extraction.text import TfidfVectorizer

        # Drop missing values and reset index
        df = df[['text', 'target']].dropna().reset_index(drop=True)

        # Split into features and labels
        X = df['text']
        y = df['target']

        # Train/test split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

        # TF-IDF Vectorization
        tfidf = TfidfVectorizer(max_features=5000)
        X_train_tfidf = tfidf.fit_transform(X_train)
        X_test_tfidf = tfidf.transform(X_test)

        X_train_tfidf.shape, X_test_tfidf.shape

Out[4]: ((4485, 5000), (1122, 5000))
```

Figure A.1: Train-test split and vectorization for Supervised ML models

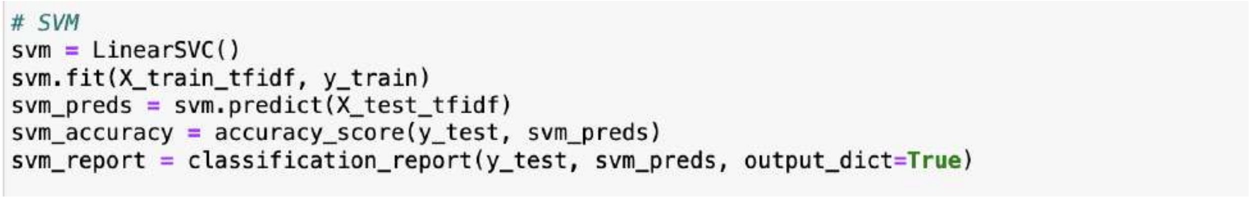
Logistic Regression Model Training



```
# Logistic Regression
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train_tfidf, y_train)
log_preds = log_reg.predict(X_test_tfidf)
log_accuracy = accuracy_score(y_test, log_preds)
log_report = classification_report(y_test, log_preds, output_dict=True)
```

Figure A.2: Logistic Regression training code

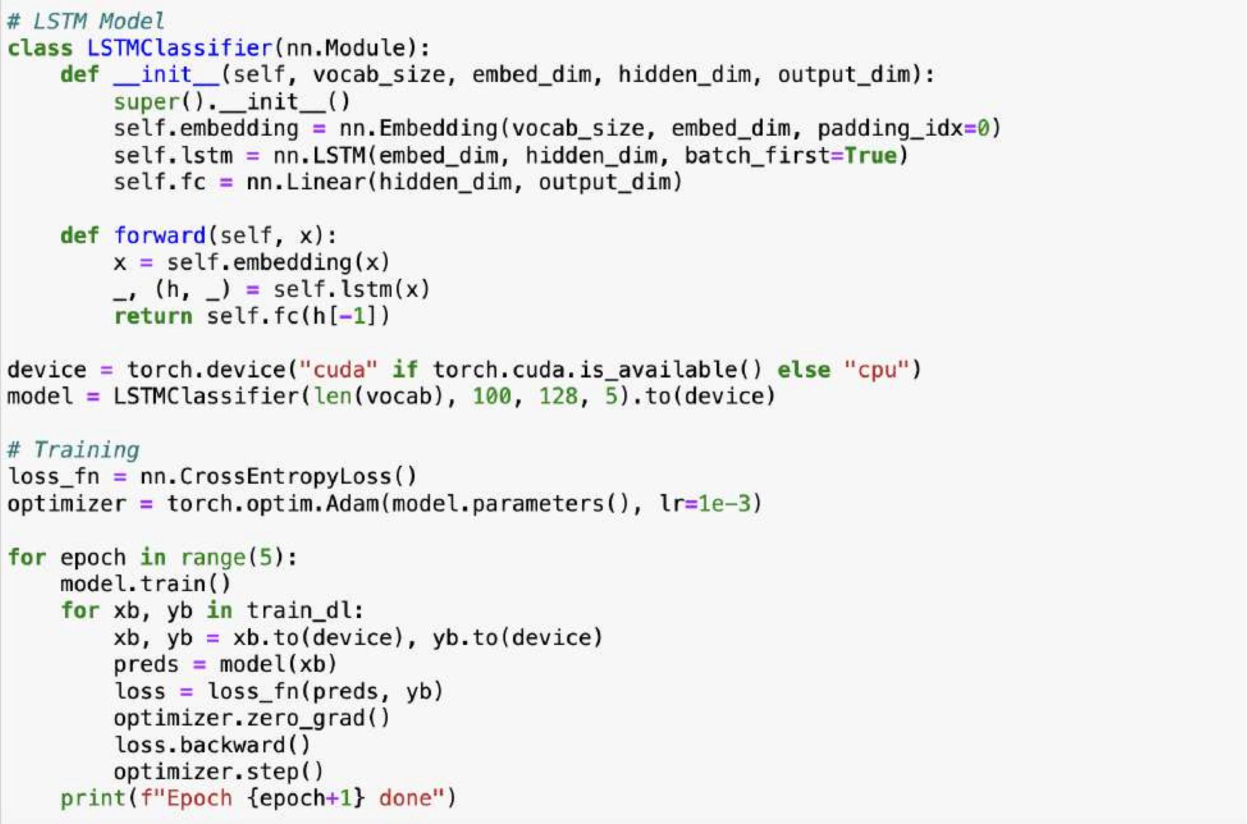
SVM Model Training



```
# SVM
svm = LinearSVC()
svm.fit(X_train_tfidf, y_train)
svm_preds = svm.predict(X_test_tfidf)
svm_accuracy = accuracy_score(y_test, svm_preds)
svm_report = classification_report(y_test, svm_preds, output_dict=True)
```

Figure A.3: SVM training code

LSTM Model Training

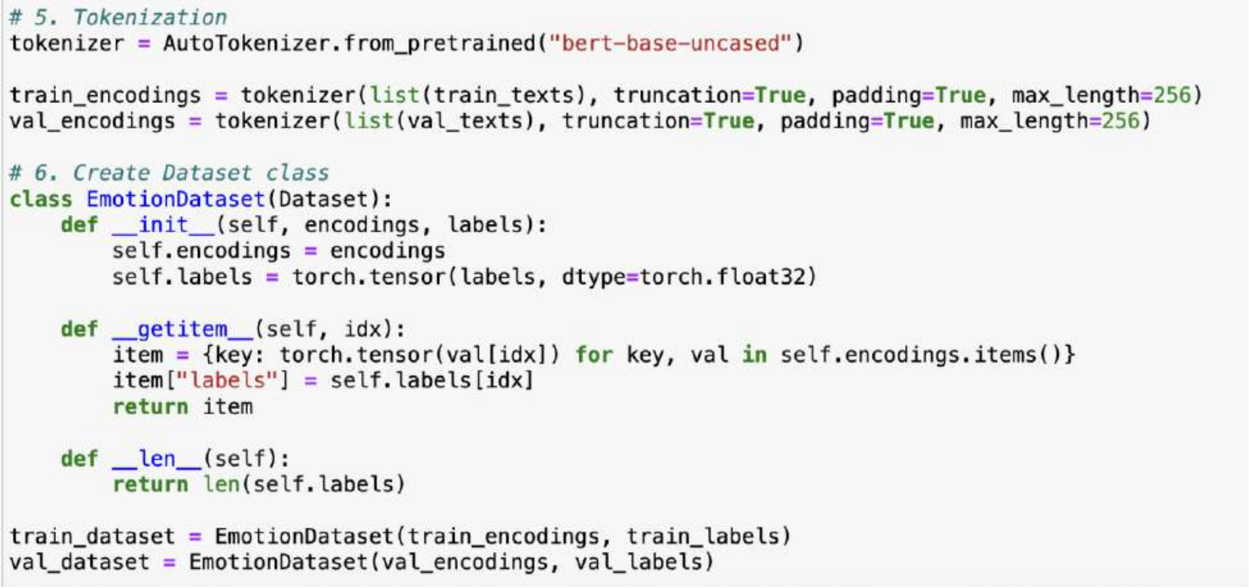


```
# LSTM Model
class LSTMClassifier(nn.Module):
    def __init__(self, vocab_size, embed_dim, hidden_dim, output_dim):
        super().__init__()
        self.embedding = nn.Embedding(vocab_size, embed_dim, padding_idx=0)
        self.lstm = nn.LSTM(embed_dim, hidden_dim, batch_first=True)
        self.fc = nn.Linear(hidden_dim, output_dim)

    def forward(self, x):
        x = self.embedding(x)
        _, (h, _) = self.lstm(x)
        return self.fc(h[-1])

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = LSTMClassifier(len(vocab), 100, 128, 5).to(device)

# Training
loss_fn = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

for epoch in range(5):
    model.train()
    for xb, yb in train_dl:
        xb, yb = xb.to(device), yb.to(device)
        preds = model(xb)
        loss = loss_fn(preds, yb)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    print(f"Epoch {epoch+1} done")
```

Figure A.4: LSTM Model Execution

APPENDIX B: BERT Model (BERT_Model.ipynb)

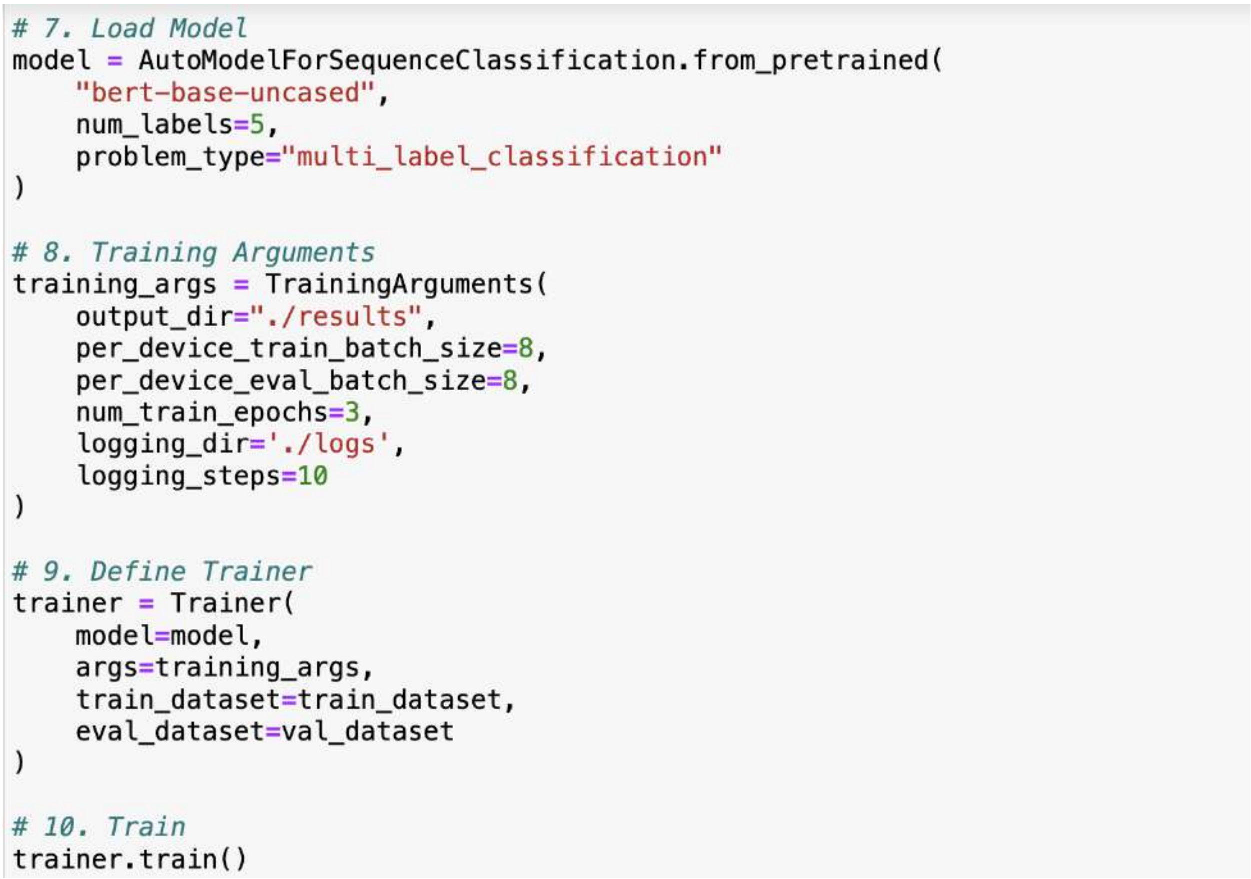
Tokenizer and Dataset Preparation



```
# 5. Tokenization
tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")

train_encodings = tokenizer(list(train_texts), truncation=True, padding=True, max_length=256)
val_encodings = tokenizer(list(val_texts), truncation=True, padding=True, max_length=256)

# 6. Create Dataset class
class EmotionDataset(Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = torch.tensor(labels, dtype=torch.float32)

    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
        item["labels"] = self.labels[idx]
        return item

    def __len__(self):
        return len(self.labels)

train_dataset = EmotionDataset(train_encodings, train_labels)
val_dataset = EmotionDataset(val_encodings, val_labels)
```

Figure B.1: BERT tokenization and dataset preparation

Model Definition and Training

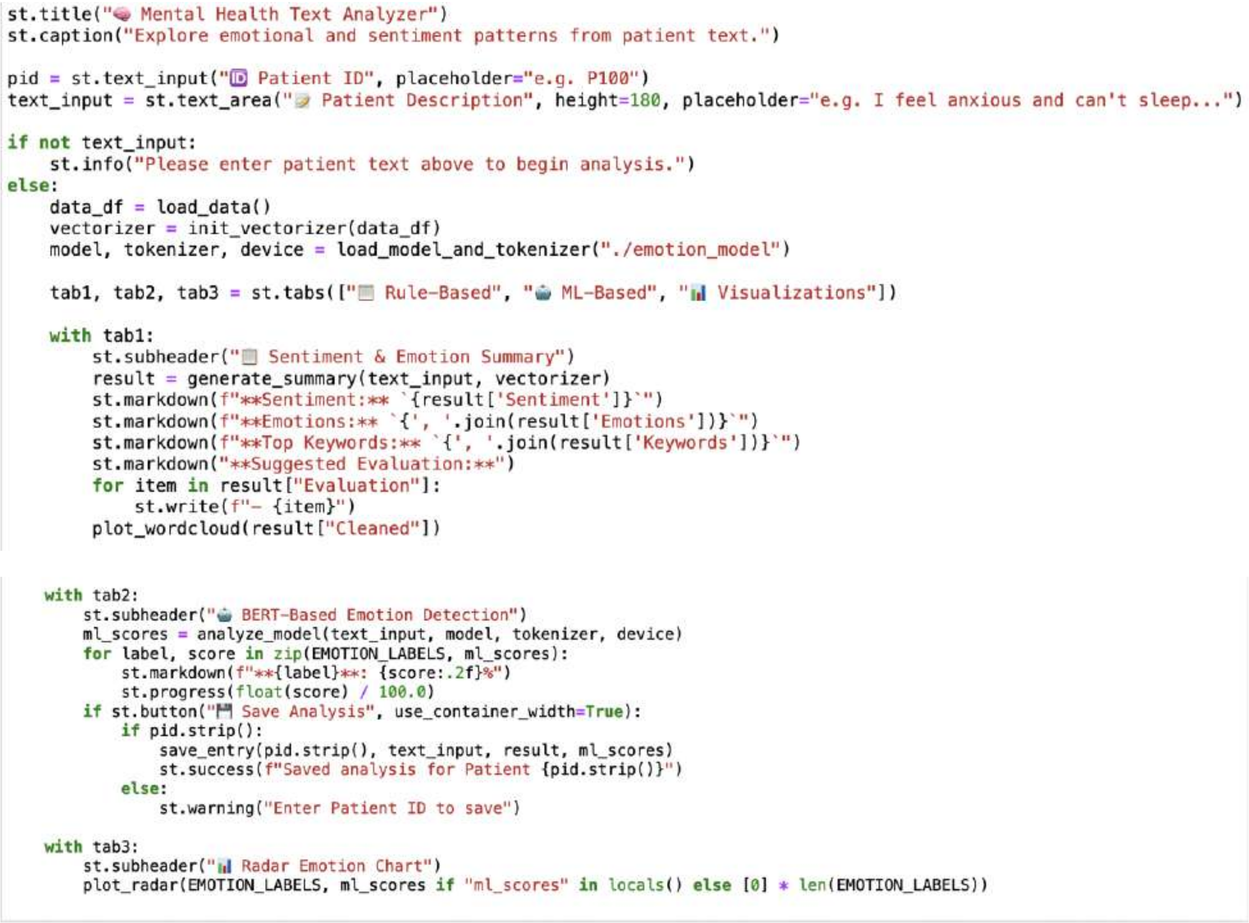


```
# 7. Load Model
model = AutoModelForSequenceClassification.from_pretrained(
    "bert-base-uncased",
    num_labels=5,
    problem_type="multi_label_classification"
)

# 8. Training Arguments
training_args = TrainingArguments(
    output_dir="./results",
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    logging_dir='./logs',
    logging_steps=10
)

# 9. Define Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset
)

# 10. Train
trainer.train()
```

Figure B.2: BERT Model Training

APPENDIX C: Deployment App (final_app.py)

Streamlit App Layout



```
st.title("🧠 Mental Health Text Analyzer")
st.caption("Explore emotional and sentiment patterns from patient text.")

pid = st.text_input("🆔 Patient ID", placeholder="e.g. P100")
text_input = st.text_area("📝 Patient Description", height=180, placeholder="e.g. I feel anxious and can't sleep...")

if not text_input:
    st.info("Please enter patient text above to begin analysis.")
else:
    data_df = load_data()
    vectorizer = init_vectorizer(data_df)
    model, tokenizer, device = load_model_and_tokenizer("./emotion_model")

    tab1, tab2, tab3 = st.tabs(["🧾 Rule-Based", "🤖 ML-Based", "📊 Visualizations"])

    with tab1:
        st.subheader("🧾 Sentiment & Emotion Summary")
        result = generate_summary(text_input, vectorizer)
        st.markdown(f"**Sentiment:** `{result['Sentiment']}`")
        st.markdown(f"**Emotions:** `{', '.join(result['Emotions'])}`")
        st.markdown(f"**Top Keywords:** `{', '.join(result['Keywords'])}`")
        st.markdown("**Suggested Evaluation:**")
        for item in result["Evaluation"]:
            st.write(f"- {item}")
        plot_wordcloud(result["Cleaned"])
```

```
    with tab2:
        st.subheader("🤖 BERT-Based Emotion Detection")
        ml_scores = analyze_model(text_input, model, tokenizer, device)
        for label, score in zip(EMOTION_LABELS, ml_scores):
            st.markdown(f"**{label}**: {score:.2f}%")
            st.progress(float(score) / 100.0)
        if st.button("💾 Save Analysis", use_container_width=True):
            if pid.strip():
                save_entry(pid.strip(), text_input, result, ml_scores)
                st.success(f"Saved analysis for Patient {pid.strip()}")
            else:
                st.warning("Enter Patient ID to save")

    with tab3:
        st.subheader("📊 Radar Emotion Chart")
        plot_radar(EMOTION_LABELS, ml_scores if "ml_scores" in locals() else [0] * len(EMOTION_LABELS))
```

Figure C. 1: Deployment App code