

# **Multimodal AI Framework for Social Media Based Mental Disorder Detection and Personalized Wellbeing Insights**

*Submitted in partial fulfillment of the  
requirements for the degree*

*of*

**Bachelor of Technology**

*by*

**SOUMYADEEP NANDY (13000121033)**

**PRITHWISH SARKAR (13000121037)**

**SAGNIK MUKHOPADHYAY (13000121040)**

**ARKAPRATIM GHOSH (13000121058)**

*June, 2025*



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
TECHNO MAIN SALT LAKE  
EM 4/1, SALT LAKE, SECTOR – V, KOLKATA – 700091**

# CERTIFICATE

This is to certify that the project entitled "**Multimodal AI Framework for Social Media Based Mental Disorder Detection and Personalized Wellbeing Insights**" prepared by **SOUMYADEEP NANDY (13000121033)**, **PRTIHWISH SARKAR (13000121037)**, **SAGNIK MUKHOPADHYAY (13000121040)** and **ARKAPRATIM GHOSH (13000121058)** of B.Tech (Computer Science & Engineering), Final Year, has been done according to the regulations of the Degree of Bachelor of Technology in Computer Science & Engineering. The candidates have fulfilled the requirements for the submission of the project report.

It is to be understood that, the undersigned does not necessarily endorse any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

---

(Signature of the Internal Guide)

---

(Signature of the HOD)

---

(Signature of the External Examiner)

.....  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
.....

## **ACKNOWLEDGEMENTS**

We would like to express our sincere gratitude to our project guide in Computer Science and Engineering department. We are extremely thankful for the keen interest he / she took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staffs for the gracious hospitality they offered us.

**SOUMYADEEP NANDY (13000121033)**

**PRITHWISH SARKAR (13000121037)**

**SAGNIK MUKHOPADHYAY (13000121040)**

**ARKAPRATIM GHOSH (13000121058)**

Techno Main Salt Lake  
Date: 21st June, 2025

# Contents

<b>Abstract</b>	1
<b>1 Introduction</b>	2
1.1 Project Overview	2
1.2 Project Purpose	2
1.3 Technical Domain Specifications	2
1.4 Business Domain Specifications	3
1.5 Glossary / Keywords	4
<b>2 Related Studies</b>	6
<b>3 Problem Definition and Preliminaries</b>	8
3.1 Context and Background	8
3.2 Objective and Challenges	8
3.3 Scope, Exclusions and Assumptions	8
<b>4 Proposed Solution</b>	9
<b>5 Project Planning</b>	10
5.1 Software Life Cycle Model	10
5.2 Dependencies, Milestones and Scheduling	11
<b>6 Requirement Analysis</b>	13
6.1 Requirement Matrix	13
6.2 Requirement Elaboration	14
<b>7 Design</b>	15
7.1 Technical Environment	15
7.2 Hierarchy of Modules	17
7.3 Detailed Design	18
<b>8 Implementation</b>	28
8.1 Data Collection and Dataset Preparation	28
8.2 Data Cleaning and Feature Extraction	30
8.3 Machine Learning Models	32
8.4 Deep Learning Models	33
8.5 Ensemble Model	38
8.6 Wellbeing Survey and Association Matrix	39
8.7 RAG for Wellbeing Insights	43
<b>9 Test Plans, Results and Analysis</b>	46
9.1 Results from Base Models	48
9.2 Comparison of different tokenizations	59
9.3 Results from Ensemble Model Training and Testing	60
9.4 Results from hierarchical Ensemble Models	66
9.5 User Interface of the Application	69
<b>10 Conclusion</b>	73
<b>11 References</b>	74

## Abstract

This project employs artificial intelligence to detect early signs of mental health issues by analyzing text, images, videos shared on platforms like Reddit and Twitter. It also allows to detect mental issues from PDFs and responses a user makes when a questionnaire is presented to him/her on the basis of an image shown. It integrates techniques such as emotion detection, image captioning, multilingual support, and psychological assessments inspired from Rorschach Inkblot Test and Ryff's Wellbeing Scale. Using an ensemble of models (Logistic Regression, SVM, Naive Bayes, LSTM, XGboost, Transformer), the system achieves 98.03% accuracy, with a hierarchical model reaching 96.25% on larger datasets. The web application supports model retraining, updating knowledge base, wellbeing surveys and providing actionable insights using Gemini and RAG (Retrieval Augmented Generation), incorporating visualizations and mapping mental health concerns to wellbeing parameters for timely interventions.

# 1 Introduction

## 1.1 Project Overview

Mental health disorders—including depression, anxiety, bipolar disorder, and PTSD—affect millions of people worldwide, creating a growing public health concern. In today's digital era, social media platforms have become a ubiquitous space where individuals freely express their emotions, share personal struggles, and document life's challenges, often without realizing that these expressions can serve as early indicators of mental health issues. This project leverages advanced machine learning and deep learning techniques to analyze social media posts by examining intricate language patterns, contextual cues, and sentiment variations to detect signs of mental distress. By employing a multimodal approach, the framework not only processes text data through methods like TF-IDF, Bag-of-Words, and deep neural networks but also integrates additional data sources such as images, videos, PDFs, and user responses to visual stimuli. This integration also allows the system to extract textual information from PDFs using OCR, analyze emotions via specialized image processing modules, and incorporate a comprehensive wellbeing survey. The survey responses are then mapped to specific parameters of Ryff's Wellbeing Scale using an association matrix, which provides personalized insights and actionable recommendations using Gemini and RAG for early intervention. Ultimately, this holistic framework is designed to facilitate timely support and guide individuals towards appropriate mental health services, contributing to a more proactive approach in managing mental health on a global scale.

## 1.2 Project Purpose

This project aims to develop a web application using machine learning and deep learning models (Logistic Regression, Naive Bayes, SVM, XGBoost, LSTM, Transformer) to detect mental health disorders from text, images, videos, PDFs, user responses to images, and social media profiles. By addressing technical challenges in processing large datasets and optimizing algorithms, the project seeks to enable early detection of mental health issues, contributing to public health improvement through technology.

## 1.3 Technical Domain Specifications

This project falls within the intersection of natural language processing (NLP) and machine learning (ML), leveraging techniques such as text vectorization, and classification algorithms. Here are the key technical domain specifications:

- **Hardware :** The project does not require specialized hardware beyond a standard machine with adequate processing power. However, for larger datasets or complex model training, a machine equipped with a GPU (Graphics Processing Unit) could significantly reduce processing time. The project can be run on any system with at least 8GB of RAM and a multi-core processor.
- **Operating System :** The project is cross-platform and can be developed and executed on any modern operating system, including macOS, Windows 10/11, Linux distributions (Ubuntu, Linux Mint).
- **Software :**
  - **Programming Languages :** Python 3.x will be the primary programming language, given its extensive libraries for machine learning, data analysis, and NLP.
  - **Libraries / Frameworks :** This project leverages a comprehensive suite of libraries and frameworks to enable its functionality, including data processing (Pandas, NumPy), machine learning (Scikit-learn, XGBoost, TensorFlow, Transformers), image and text analysis (OpenCV, Tesseract, Pytesseract, DeepFace), audio processing (Librosa, PyDub, SpeechRecognition), social media integration (PRAW, Tweepy), and visualization (Plotly, Matplotlib). Additional tools like Streamlit, NLTK, and Google Generative AI ensure seamless application development, multilingual support, and advanced model implementation.
  - **Development Environment :** Google Colab is used for cloud-based execution when working with larger datasets or GPU-based model training.

## 1.4 Business Domain Specifications

From a business perspective, this project offers significant value in mental health monitoring, public health awareness, and social media governance. As mental health issues rise globally, organizations seek innovative solutions to address this crisis. By using machine learning to detect mental health disorders from social media data, this project can transform mental health management at both individual and societal levels, impacting various industries effectively.

- **Mental Health Services :** Mental health providers, including hospitals and therapy centers, can leverage machine learning to detect early signs of mental disorders from social media data. This proactive approach complements traditional self-reporting and clinical assessments, enabling earlier intervention and support for patients.

- **Social Media Platforms** : Social media platforms like Twitter and Reddit are key spaces for expressing thoughts and emotions, including mental health struggles. This project's machine learning models can help these platforms safeguard user well-being by identifying concerns early, while maintaining ethical standards.
- **Public Health Organizations** : Public health organizations can use real-time social media data to monitor mental well-being, identify trends, and design data-driven interventions. By analyzing language patterns, they can create targeted awareness campaigns that better engage individuals facing mental health challenges.

## 1.5 Glossary / Keywords

Term	Definition
Natural Language Processing (NLP)	A branch of artificial intelligence focused on the interaction between computers and humans through natural language, including tasks like text analysis.
Retrieval-Augmented Generation (RAG)	A hybrid NLP framework that combines information retrieval and text generation by fetching relevant context from a knowledge base before generating responses, improving factual accuracy and relevance.
Vectorization	The process of converting textual data into numerical form (such as a vector) so that it can be used as input for machine learning models.
Classifier	A machine learning model or algorithm that categorizes or labels data points into predefined classes.
Mental Health Disorder	A wide range of conditions that affect mood, thinking, and behavior, including depression, anxiety, schizophrenia, etc.
Data Preprocessing	The process of preparing raw data for analysis by cleaning, normalizing, and transforming it into a usable format for machine learning models.
Cross-validation	A model validation technique used to assess how well a model performs by dividing data into training and testing sets multiple times for better accuracy.
Precision	In the context of classification, precision refers to the accuracy of positive predictions, calculated as the ratio of true positives to the sum of true and false positives.
Recall	In classification, recall measures the ability of a model to identify all relevant instances within a dataset, calculated as the ratio of true positives to the sum of true positives and false negatives.
PRAW	PRAW (Python Reddit API Wrapper) is a Python library that provides a simple interface to interact with Reddit's API for accessing Reddit data, such as posts, comments, and user information.

<b>Term</b>	<b>Definition</b>
TesseractOCR	TesseractOCR is an open-source Optical Character Recognition (OCR) engine that extracts text from images with high accuracy; it is widely used for various applications like scanning documents and digitalizing printed text.
Depression	There is a difference between depression and mood swings or short-lived emotional reactions to daily experiments; A mental state causing painful symptoms adversely disrupts normal activities (e.g., sleeping).
Anxiety	Several behavioral disturbances are associated with anxiety disorders, including excessive fear and worry. Severe symptoms cause significant impairment in functioning cause considerable distress. Anxiety disorders come in many forms, such as social anxiety, generalized anxiety, panic, etc.
Bipolar Disorder	An alternating pattern of depression and manic symptoms is associated with bipolar disorder. An individual experiencing a depressive episode may feel sad, irritable, empty, or lose interest in daily activities. Emotions of euphoria or irritability, excessive energy, and increased talkativeness can all be signs of manic depression. Increased self-esteem, decreased sleep need, disorientation, and reckless behavior may also be signs of manic depression.
Post-Traumatic Stress Disorder (PTSD)	In PTSD, persistent mental and emotional stress can occur after an injury or severe psychological shock, characterized by sleep disturbances, constant vivid memories, and dulled response to others and the outside world. People who re-experience symptoms may have difficulties with their everyday routines and experience significant impairment in their performance.
DeepFace	DeepFace is a Python library for deep learning-based facial recognition and attribute analysis. It supports several pre-trained models and simplifies face recognition tasks, making it suitable for various applications in image analysis.
Transformers Module	The Transformers module in Python, developed by Hugging Face, is a library for natural language processing (NLP) tasks like text classification, translation, and summarization, using state-of-the-art models like BERT and GPT.
Gemini 2.0 Flash	Gemini 2.0 Flash is a cutting-edge AI model developed by Google, capable of performing advanced generative and analytical tasks across text, image, and other modalities.
FFmpeg	FFmpeg is a multimedia framework used for encoding, decoding, transcoding, streaming, and manipulating audio and video files, supporting a wide range of formats and codecs.
Hyperparameter Tuning	Hyperparameter tuning involves selecting the best parameters for a machine learning model to optimize its performance on a given task, using methods like grid search or random search.

## 2 Related Studies

Study	Summary
Choudhury et al. (2013)	Explored the predictive capabilities of social media content in identifying depression by analyzing Twitter data. They discovered that specific linguistic patterns (e.g., negative emotion words) correlated strongly with self-reported depressive symptoms [2].
Guntuku et al. (2017)	Conducted an integrative review which synthesized various methodologies highlighted that social media platforms are rich sources of data, revealing critical information about users' mental health [4].
Mathur et al. (2022)	Provided a systematic review analysing machine learning techniques for mental health detection using social media data, leveraging both individual assessments and broader epidemiological studies [5].
Nadeem (2016)	Investigated depression identification on Twitter by developing algorithms to discern emotional cues in tweets revealing that simple text analysis could lead to improvements in identifying mental risks [6].
AlSagri and Ykhlef (2020)	Introduced a machine learning-based approach for depression detection on Twitter that combined both content and activity features. Their work demonstrated that a fusion of linguistic and behavioral analysis can enhance the accuracy of depression identification [1].
Vaishnavi et al. (2022)	Examined various machine learning algorithms for predicting mental health illnesses using social media posts. Their findings emphasized that certain algorithms outperform others in classifying mental health conditions, underlining the importance of algorithm selection [9].
Safa et al. (2023)	Presented a roadmap for predicting mental health using social media, highlighting ongoing challenges such as ethical considerations and data privacy. They stressed the need for a robust ethical framework in research that leverages social media data [7].
Ensemble learning using transformers for NLP	Provided a comprehensive review of transformer models (BERT, XLNet, RoBERTa, GPT-2, ALBERT) across multiple NLP tasks. The study introduced ensemble learning with these models and demonstrated that ensemble approaches can significantly improve performance over single classifier methods [10].
Ensemble hybrid model for depression detection	Proposed an ensemble hybrid model combining SVM and MLP to improve depression prediction accuracy. Addressing class imbalance with SMOTE and cluster sampling, the model achieved an accuracy of 99.39% and an F1-score of 99.51%, outperforming previous approaches [8].
Single classifier vs. ensemble ML approaches	Explored various machine learning techniques to predict mental health issues using survey responses from OSMI. The study compared single classifiers (e.g., Logistic Regression, Gradient Boosting, Neural Networks, KNN, SVM) with ensemble approaches, finding that Gradient Boosting achieved the highest accuracy (88.80%), followed by Neural Networks (88.00%), and that the ensemble classifier achieved 85.60% accuracy [3].

## Preframe Study Review

Sl No.	Paper Title and Author	Year	Aim and Objective	Uniqueness Claimed	Outcome Achieved	Algorithm/Tools/Platform	Feature Extraction		Modality of Data	Data Resource
1	Combining Sentiment Analysis Models Using Stacking Ensemble Learning Techniques on BIST30 Stocks by Mahmut Sami SIVRI	2024	Develop a stacking ensemble model to enhance sentiment analysis accuracy and robustness for BIST30 financial news.	Integrate diverse models (LSTM, BERT, Naive Bayes, SVM) in a stacking ensemble with a focus on BIST30 stocks, robust data strategies, and comparative analysis.	Enhance financial sentiment analysis with superior performance, generalization over baseline models with accuracy of 85%	Stacking ensemble with LSTM, BERT, Naive Bayes, SVM, and Logistic Regression, trained on 5,287 financial news articles (2020-2023) with an 80-10-10 split.	SVM uses TF-IDF with text preprocessing; cleaning, tokenization, stopwords removal, and lemmatization.	Financial news articles classified into Positive, Negative, and Neutral sentiment categories.	Thomson Reuters news articles	
2	Predicting mental health using social media: A roadmap for future development by Ramin Safa, S. A. Edalatpanah, Ali Sorourkhah	2022	A roadmap for analyzing and predicting mental disorders using social media data, covering collection, feature extraction, and prediction methods.	Comprehensive analysis of mental state assessment methods on social data, categorizing approaches and discussing challenges and future directions.	Conceptual framework for environmental health research via social media, covering assessment strategies, data methods, algorithms, metrics, and future challenges.	Machine learning, NLP, and sentiment analysis using platforms like Twiter and Facebook, with tools like LIWC and frameworks such as CNN and LSTM.	Tools like LIWC, OpinionFinder, SentStrength, VADER, Word2Vec, LDA, tf-idf, VGG-Net and ImageNet.	Textual Data and Visual Content	myPersonality project CLPsych workshops data, AutoDep dataset, SDCNL dataset, CAMS dataset	
3	Detecting Depression and Mental Illness on Social Media: An Integrative Review by Sharath Chandra Gunuku, David B. Yaden, Margaret L. Kern, Lyle H. Ungar, Johannes C. Eichstaedt	2016	Reviewing recent studies and comparing approaches for predicting mental illness through social media analysis.	Comprehensive review comparing mental illness detection methods and prediction performances with clinical baselines.	Social media-based screening shows AUCs between 0.70 and 0.91, bridging clinician assessment and screening surveys with accuracy of 87%.	Using Linear Regression, SVM, Neural Networks, and Random Forest with analysis tools like LIWC and LabMT.	LIWC and N-Gram	Textual Data	Survey and Social media	
4	Single classifier vs. ensemble machine learning approaches for mental health prediction by Jeti Chung and Jason Teo	2023	Empirically evaluate and compare single classifiers and ensemble approaches for predicting mental health problems.	Comprehensive comparison of traditional algorithms, Deep Neural Networks, XGBoost, and ensemble methods for mental health prediction using survey data.	Gradient Boosting led with 88.80% accuracy, followed by Neural Networks (88.00%) and XGBoost (87.20%).	Logistic Regression, Gradient Boosting, Neural Networks, K-NN, SVM, XGBoost, and Ensemble Voting Classifier.	Extra Trees Classifier was used for feature selection to reduce overfitting	Survey	OSIM's 2014 Mental Health in Tech Survey	
5	Survey of transformers and towards ensemble learning using transformers for natural language processing by Hongzhi Zhang and M. Omair Shafiq	2024	Compare and analyze transformer models across six NLP tasks, developing ensemble models to leverage their strengths.	Outperformed single classifiers with two ensemble models, identifying optimal combinations for specific NLP tasks with accuracy of 79%.	Using transformer embeddings, word vectors, BERTopic, TF-IDF, and contextual semantic representations for NLP.	Using transformer-based embeddings, word vectors, BERTopic for topic modeling, class-based TF-IDF, and contextual semantic representations.	Using datasets like Kaggle's coronavirus tweets, SQAD1.1, Groningen Meaning Bank, CNN daily mail, disaster tweets, and Trump 2020 election speeches for various NLP tasks.			
6	Ensemble of hybrid model based technique for early detecting of depression based on SVM and neural networks by Dip Kumar Saha, Tuhin Hosain, Mejjl Sofian, Sultan Alfarhan, M. F. Mirtha & Darren Che	2024	Develop an automated system to detect depression using psychological and socio-demographic characteristics.	Developed an automated system with improved accuracy and reduced class imbalance with accuracy of 99.39%.	Algorithms like SVM, MLP, Hybrid DapMVM, RFC, KNN, GNB with SMOTE, cluster sampling, and Scikit-learn for depression detection.	Using Label Encoder, Standard Scaler and Feature selection techniques for data preprocessing.	Survey data with 30 psychosocial/demographic predictors and 1 depression status target, including 25 BDI questions.			
7	Machine Learning-based Approach for Depression Detection in Twitter Using Content and Activity Features by Haoot AlSagri and Mourad Ykhlaf	2023	Detect depression in Twitter users by analyzing tweet content and network behavior using machine learning techniques.	Incorporating tweet text features like activity categories, Depi_Sent lexicon, and synonyms for enhanced depression detection.	Developed a binary classification model with improved accuracy by combining user activities and tweets for better mental health detection with accuracy 82.5%.	Using SVM, Naive Bayes, and Decision Tree algorithms with R 3.3, RSudio, Twitter API, WordNet, and R packages for data analysis.	Text data (tweets, pronouns, sentiment words, depression terms) and user activity data (followers, posts, mentions, retweets, emojis, hashtags, replies).			
8	Predicting Depression via Social Media' Authors: Munmun De Choudhury, Michael Gamon, Scott Counts, Eric Horvitz	2013	Leveraging social media behavioral patterns to detect and predict Major Depressive Disorder (MDD) early.	Pioneering crowdsourced depression prediction.	Achieved 70% accurate depression prediction, highlighting key behavioral and linguistic markers.	Utilized SVM with 10-BF kernel, PCA, and 10-fold cross-validation on Twitter data for depression analysis.	Text data (tweets, demographics, and social network data for depression detection).			
9	Predicting Mental Health Illness using Machine Learning Algorithms by Konda Vaishnavi	2022	Evaluate and compare machine learning techniques for accurate mental health issue prediction.	Unique comparison of five ML techniques for mental health prediction using diverse accuracy metrics and ROC curves.	Achieved 81.75% accuracy with stacking, with all classifiers showing strong ROC values (0.8-0.9).	Utilized Logistic Regression, K-NN, Decision Stacking for mental health prediction.	Used a dataset with 27 attributes and 1259 entries, including text documents for mental health prediction.			
10	Generalizability of Machine Learning to Categorize Various Mental Illness Using Social Media Activity Patterns by Ang, C.S. & Venkatachala, R.	2023	Explore linguistic patterns and cross-platform machine learning models for classifying mental health groups based on social media activity.	Improved accuracy by 2.11% with simpler cross-platform models for mental health classification on Twitter and Reddit.	Achieved 91.5% accuracy with better Reddit model generalization and distinct linguistic patterns between platforms.	Employed LIWC, Porter Stemmer, and Google Colab for Twitter and Reddit-based mental health classification.	Used 23M Twitter posts and 606K Reddit posts from mental health conditions from relevant subreddits and hashtags.			

### 3 Problem Definition and Preliminaries

#### 3.1 Context and Background

Mental health disorders affect approximately 1 in 8 people globally. Social media platforms like Reddit and Twitter offer vast amounts of real-time data reflecting mental health struggles, but the unstructured nature of this data poses challenges for effective identification and categorization of specific disorders.

#### 3.2 Objective and Challenges

The primary objective is to develop a system that uses NLP and machine learning to analyze Reddit and Twitter posts for detecting mental health disorders like depression, anxiety, bipolar disorder, and PTSD. It seeks to **classify posts** accurately and provide **data-driven insights** into mental health trends for researchers, professionals, and policymakers.

- **Data Variability:** Social media posts vary in structure, style, and language.
- **Imbalanced Data:** Uneven distribution of mental issues can impact model training.
- **Cultural Nuances:** Mental health discussions differ across cultures.
- **Privacy and Ethics:** Analyzing social media data raises concerns about user privacy.

#### 3.3 Scope, Exclusions and Assumptions

The scope of the project is to develop a multimodal AI framework to detect mental disorders from social media inputs using ML and NLP. It analyzes text, images, videos, PDFs, dynamic responses to image shown and Reddit/Twitter data via APIs, classifying inputs into Normal, Anxiety, Depression, Bipolar, or PTSD using a Reddit dataset. Finally an association is created between mental health disorder and mental wellbeing parameters from Ryff's Psychological Well-being Scale.

This project excludes real-time sentiment analysis, platform-specific features like hashtags or subreddits, and ethical implications of data ownership. It also does not analyze comments, metadata, or Reddit/Twitter-specific elements, focusing solely on detecting mental health disorders from user profiles and mapping them to wellbeing insights, though the dataset has been made solely from Reddit posts from various subreddits.

The project assumes that the Reddit dataset obtained via PRAW represents diverse mental health discussions and that user posts accurately reflect emotions. NLP techniques are presumed effective for sentiment classification, and selected ML models (Logistic Regression, SVM,

Naïve Bayes, LSTM, Transformer, XGBoost) are expected to perform optimally. Social media sentiments are considered valid proxies for public mental health perceptions. Data preprocessing is assumed sufficient to reduce noise, and ethical standards are maintained to protect user privacy. The users' responses to the well-being surveys are considered valid for updating the association matrix between mental health disorders and well-being parameters.

## 4 Proposed Solution

### Special Contributions

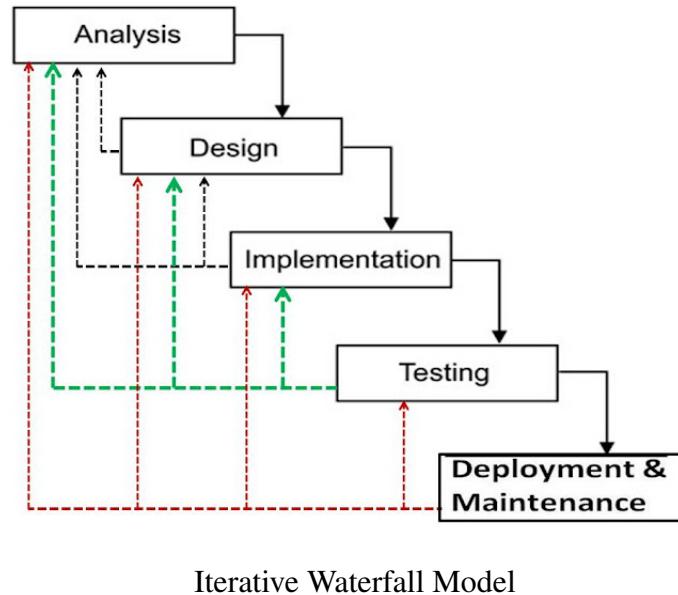
Component	Description
Dataset Acquisition	Reddit data was sourced using PRAW from relevant subreddits (Normal, Depression, Anxiety, Bipolar, PTSD). Extensive preprocessing ensured data integrity.
Text Vectorization	TF-IDF and Bag-of-Words were used to convert text into numerical format via Scikit-learn, enabling efficient feature extraction and model training. Other vectorization techniques like Word2Vec, LIWC, and N-Grams were also explored.
Machine Learning Models	Logistic Regression, SVM, Naïve Bayes, LSTM, Transformer, and XGBoost were implemented for multi-class classification of mental health conditions.
Model Evaluation	Accuracy, precision, recall, and F1-score were used to assess performance. Confusion matrices and ROC curves were generated for detailed model evaluation.
Insights & Recommendations	Findings inform mental health professionals and policymakers on probable issues and provide well-being insights.
Documentation & Reproducibility	Detailed documentation ensures usability, including methodology, code, and instructions for result reproduction.

### Reusable Components

Module / Function	Description
Data Collection Functions	Modular functions designed for data collection, which can be reused across different platforms.
Data Preprocessing Module	A component that cleans data by removing duplicates and empty rows, and adds a separate column for cleaned texts.
Machine and Deep Learning Model Functions	Functions for implementing Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, XGBoost, Long Short Term Memory, and Transformer algorithms. These functions enable easy retraining on varying datasets and feature various evaluation metrics to assess model performance.
Deployment Function	A separate function that contains the main Python file for creating a web-based application on Streamlit Cloud, including the necessary requirements and package dependencies for deployment.

## 5 Project Planning

### 5.1 Software Life Cycle Model



Project Phases and Descriptions

Phase	Description
Requirement Gathering and Analysis	This initial phase involved understanding the project's goals, objectives, and stakeholder expectations.
Data Collection and Preparation	Utilizing the Reddit API, the data collection phase was executed. This included downloading the dataset, examining its structure, and performing data cleaning and preprocessing to ensure its suitability for analysis.
Model Development	This phase included the creation of a Bag-of-Words model, splitting the dataset into training and test sets, and implementing various machine learning algorithms.
Model Evaluation	Following model development, testing and validation of the models were performed to ensure they met the required accuracy benchmarks. Performance metrics such as accuracy, precision, recall, and F1-score were used to evaluate model effectiveness.
Final Deployment and Documentation	The last phase focused on deploying the best-performing model and creating comprehensive documentation. This included user manuals and technical documentation to facilitate future maintenance and enhancements.

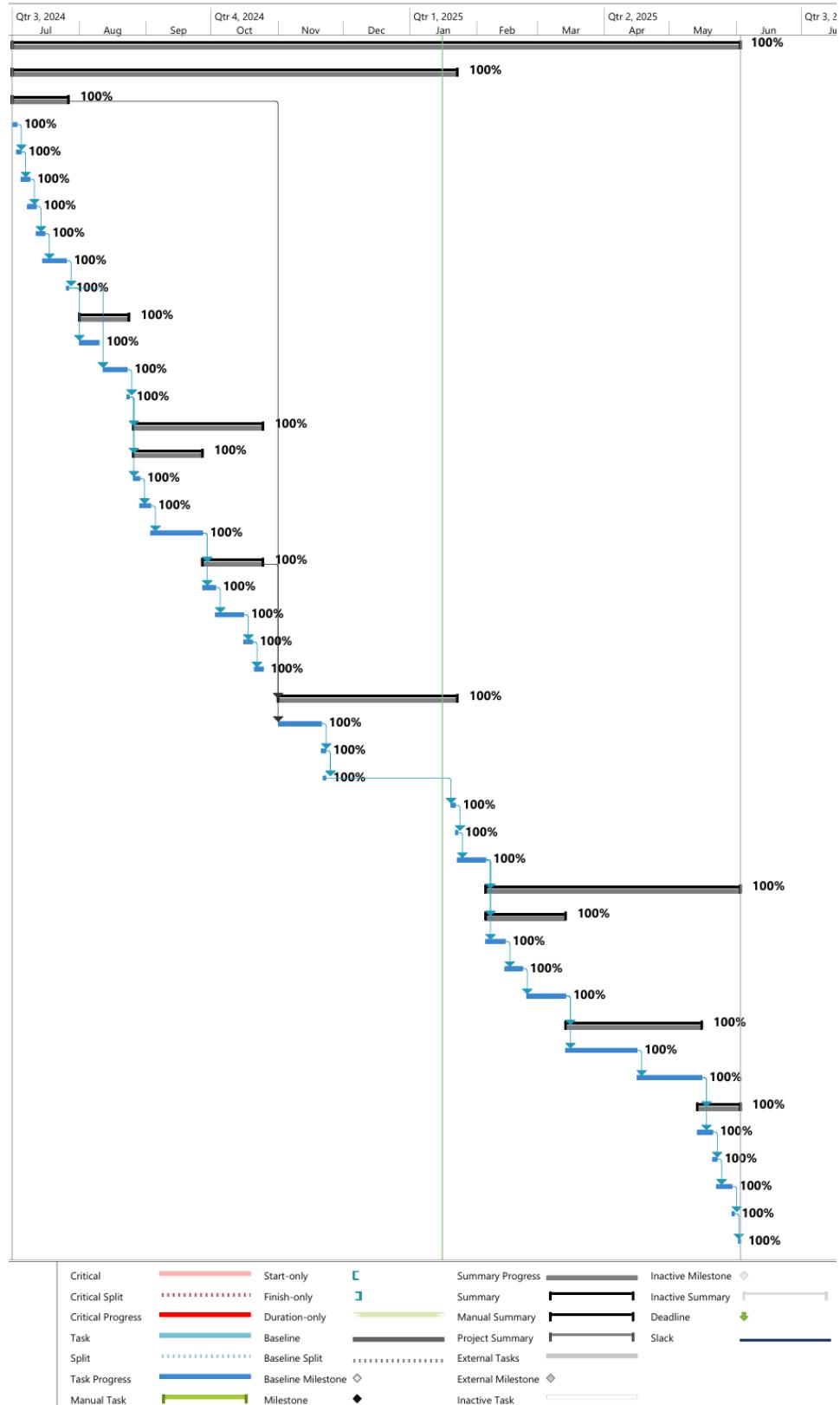
## 5.2 Dependencies, Milestones and Scheduling

Key dependencies were identified for successful project progression. For instance, completion of the data preparation phase was critical before proceeding to model development. Milestones were established at the end of each phase to ensure accountability and track progress. The successful completion of the requirement gathering phase marked the first milestone, followed by the data preparation phase, and so on. Effective scheduling is vital for project success. A detailed timeline with tasks like requirement gathering, data preprocessing, model implementation, and testing was created, with flexibility for adjustments based on feedback. Key milestones, including data analysis, model validation, and user acceptance testing, ensure progress tracking. Using tools like Microsoft Project, we monitor tasks, manage resources, and maintain communication to deliver a high-quality solution on time.

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	% Complete
1	✓	<b>GR29 MAFSMBMDDPW</b>	243 days	Mon 01-07-24	Mon 02-06-25		100%
2	✓	<b>Phase 1: 7th Semester Activities</b>	150 days	Mon 01-07-24	Wed 22-01-25		100%
3	✓	<b>Project Startup</b>	20 days	Mon 01-07-24	Fri 26-07-24		100%
4	✓	Team Building	2 days	Mon 01-07-24	Tue 02-07-24		100%
5	✓	Brainstorm on Project Topic	2 days	Wed 03-07-24	Thu 04-07-24	4	100%
6	✓	Project agreed with Guide	2 days	Fri 05-07-24	Mon 08-07-24	5	100%
7	✓	Related Study & Documentation	4 days	Mon 08-07-24	Thu 11-07-24	6	100%
8	✓	Deliver Project Synopsis for Guide's review	2 days	Fri 12-07-24	Mon 15-07-24	7	100%
9	✓	Close review feedbacks	9 days	Mon 15-07-24	Thu 25-07-24	8	100%
10	✓	<b>Project Synopsis Finalized</b>	1 day	<b>Fri 26-07-24</b>	<b>Fri 26-07-24</b>	9	<b>100%</b>
11	✓	<b>Requirement Analysis</b>	17 days	Thu 01-08-24	Fri 23-08-24		100%
12	✓	Gather Requirements	7 days	Thu 01-08-24	Fri 09-08-24	10	100%
13	✓	Prepare Draft Requirement Matrix	9 days	Mon 12-08-24	Thu 22-08-24	10	100%
14	✓	<b>Requirement Matrix Finalized</b>	1 day	<b>Fri 23-08-24</b>	<b>Fri 23-08-24</b>	13	<b>100%</b>
15	✓	<b>Design</b>	46 days	Mon 26-08-24	Thu 24-10-24	14	100%
16	✓	Detailed Design	25 days	Mon 26-08-24	Thu 26-09-24	14	100%
17	✓	Data Collection	3 days	Mon 26-08-24	Wed 28-08-24	14	100%
18	✓	Data Preprocessing	4 days	Thu 29-08-24	Mon 02-09-24	17	100%
19	✓	Model Training and Evaluation	18 days	Tue 03-09-24	Thu 26-09-24	18	100%
20	✓	<b>Test Plan Preparation</b>	21 days	<b>Fri 27-09-24</b>	<b>Thu 24-10-24</b>	19	<b>100%</b>
21	✓	Text Classification	4 days	Fri 27-09-24	Wed 02-10-24	19	100%
22	✓	Image Classification	9 days	Thu 03-10-24	Tue 15-10-24	21	100%
23	✓	Video Classification	4 days	Wed 16-10-24	Sat 19-10-24	22	100%
24	✓	Reddit and Twitter User Analysis	4 days	Mon 21-10-24	Thu 24-10-24	23	100%
25	✓	<b>Phase 1 Closure</b>	59 days	<b>Fri 01-11-24</b>	<b>Wed 22-01-25</b>	3	<b>100%</b>
26	✓	Prepare 7th Semester Project Report	14 days	Fri 01-11-24	Wed 20-11-24	20	100%
27	✓	<b>Updated Requirement Matrix</b>	2 days	<b>Thu 21-11-24</b>	<b>Fri 22-11-24</b>	26	<b>100%</b>
28	✓	<b>Updated Project Plan</b>	1 day	<b>Fri 22-11-24</b>	<b>Fri 22-11-24</b>	27	<b>100%</b>
29	✓	Project Viva	2 days	Mon 20-01-25	Tue 21-01-25	28	100%
30	✓	<b>Approved Project Report - 7th Semester</b>	1 day	<b>Wed 22-01-25</b>	<b>Wed 22-01-25</b>	29	<b>100%</b>
31	✓	<b>Semester Gap</b>	9 days	Thu 23-01-25	Tue 04-02-25	30	100%
32	✓	<b>Phase 2: 8th Semester Activities</b>	84 days	<b>Wed 05-02-25</b>	<b>Mon 02-06-25</b>	31	100%
33	✓	<b>Coding &amp; Unit Testing</b>	27 days	<b>Wed 05-02-25</b>	<b>Thu 13-03-25</b>	31	100%
34	✓	Data Collection and Preprocessing	7 days	Wed 05-02-25	Thu 13-02-25	31	100%
35	✓	Model Training and Evaluation	6 days	Fri 14-02-25	Fri 21-02-25	34	100%
36	✓	Web Application Components	14 days	Mon 24-02-25	Thu 13-03-25	35	100%
37	✓	<b>System Integration Testing</b>	45 days	<b>Fri 14-03-25</b>	<b>Thu 15-05-25</b>	36	<b>100%</b>
38	✓	API Calls	23 days	Fri 14-03-25	Tue 15-04-25	36	100%
39	✓	Deployment	22 days	Wed 16-04-25	Thu 15-05-25	38	100%
40	✓	<b>Project Closure</b>	14 days	<b>Wed 14-05-25</b>	<b>Mon 02-06-25</b>	39	<b>100%</b>
41	✓	Prepare 8th Semester Project Report	5 days	Wed 14-05-25	Tue 20-05-25	39	100%
42	✓	<b>Updated Requirement Matrix</b>	2 days	Wed 21-05-25	Thu 22-05-25	41	100%
43	✓	<b>Updated Project Plan</b>	5 days	Fri 23-05-25	Thu 29-05-25	42	100%
44	✓	Review by Faculties	1 day	Fri 30-05-25	Fri 30-05-25	43	100%
45	✓	<b>Approved Project Report - 8th Semester</b>	1 day	Mon 02-06-25	Mon 02-06-25	44	100%

### Project Plan

# MAFSMBMDDPW



Gantt Chart

## 6 Requirement Analysis

### 6.1 Requirement Matrix

Rqmt ID	Requirement Item	Requirement Analysis Status	Design Module (As per Prototype folder structure)	Design Reference (section# under project Report)
FR-001	Collect social media data from Reddit.	Completed	D01	8.2.1
FR-002	Implement data cleaning and preprocessing.	Completed	D02	8.2.2
FR-003	Train machine learning and deep learning models.	Completed	D03	8.2.3 - 8.2.10
FR-004	Evaluate models using performance metrics (accuracy, recall, F1 Score, Support).	Completed	D03	9.2 - 9.9
FR-005	Text Analysis	Completed	D04	APPENDIX A - PROTOTYPE
FR-006	Image Upload Analysis	Completed	D04	APPENDIX A - PROTOTYPE
FR-007	Video Upload Analysis	Completed	D04	APPENDIX A - PROTOTYPE
FR-008	PDF Upload Analysis	Completed	D04	APPENDIX A - PROTOTYPE
FR-009	User response to image	Completed	D04	APPENDIX A - PROTOTYPE
FR-010	Reddit and Twitter Username Analysis	Completed	D04	APPENDIX A - PROTOTYPE
FR-011	Wellbeing survey and mapping using association matrix	Completed	D04	APPENDIX A - PROTOTYPE
FR-012	Application Deployment and Model Retraining	Completed	D05	APPENDIX A - PROTOTYPE
NFR-001	Scalability and Performance	Completed	D03	10 - 11

Requirement Matrix

## 6.2 Requirement Elaboration

No	Requirement	Input	Output
FR-001	Collect social media data from Reddit	API queries	Raw text data
FR-002	Implement data cleaning and preprocessing	Raw text data	Cleaned, structured text
FR-003	Train machine learning and deep learning models	Preprocessed data	Trained models
FR-004	Evaluate models using performance metrics	Trained models	Accuracy, Recall, F1 Score
FR-005	Text Analysis	User text input	Mental disorder classification
FR-006	Image Upload Analysis	Uploaded image	Extracted text, emotions for classification
FR-007	Video Upload Analysis	Uploaded video	Extracted frames, emotions and text for classification
FR-008	PDF Upload Analysis	Uploaded PDF	Extracted text and analysis
FR-009	User response to image	User input	Mental Disorder classification based on input text
FR-010	Reddit and Twitter Username Analysis	Username input	Mental disorder trends across the top posts
FR-011	Wellbeing survey and mapping using association matrix	Survey responses	Mental health insights
FR-012	Application Deployment and Model Retraining	Updated dataset	Improved/updated model accuracy within the web application
NFR-001	Scalability and Performance	Handling bigger dataset and subset models for a global ensemble model	Maintaining overall accuracy and response time

### Functional and Non-Functional Requirements

## 7 Design

### 7.1 Technical Environment

The technical environment for the project "Multimodal AI Framework for Social Media Based Mental Disorder Detection and Personalized Wellbeing Insights" comprises a combination of hardware, software, and tools that enable smooth data analysis, machine learning model training, and deployment. Below is an overview of the minimum hardware configuration, software tools, and package details necessary to carry out this project effectively.

Component	Specification
Processor	Intel Core i5 (or equivalent) with a base clock speed of at least 2.5 GHz. A multi-core processor is preferred for parallel processing, essential for model training and data preprocessing.
RAM	8 GB recommended for handling data loading, cleaning, and transformation. For large datasets, 16 GB is ideal to prevent memory overflow and processing delays.
Storage	Minimum 256 GB SSD recommended. SSD offers faster read/write speeds, significantly improving dataset loading times, especially for large datasets like Reddit-based social media posts.
GPU	Not necessary for basic ML tasks like Logistic Regression or SVM. For deep learning, an NVIDIA GTX 1060 with 4 GB VRAM or higher is advantageous.
Operating System	Windows 10 (64-bit) or higher, macOS 10.13 (High Sierra) or higher, or any stable Linux distribution (e.g., Ubuntu 18.04 or higher). The OS should support ML libraries and project tools.

#### Minimum Hardware Configuration

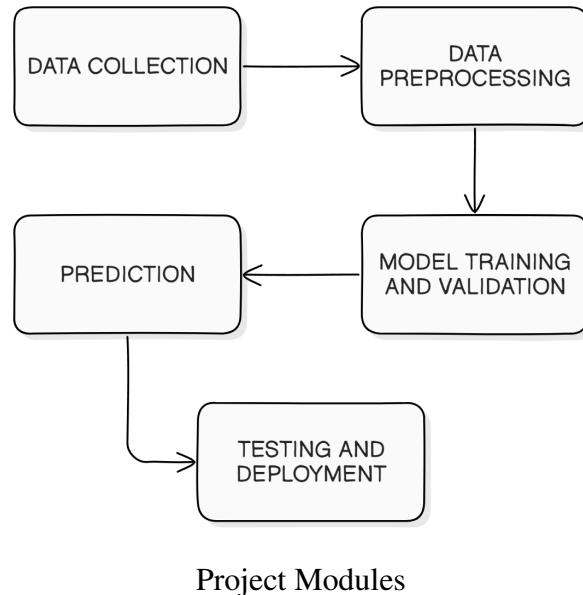
Library/Package	Description
Python	Primary programming language for data processing and model training.
pandas	Data manipulation and preprocessing.
scikit-learn	Machine learning models and evaluation tools.
Streamlit	Web framework for deploying interactive ML applications.
pyngrok	Creates secure tunnels for sharing local applications.
Google Colab	Cloud-based Python environment with GPU/TPU support.
PRAW	Access and retrieve Reddit data via API.
pytesseract	OCR library for extracting text from images.
Pillow	Image processing and handling library.

#### Libraries and Packages Used

<b>Library/Package</b>	<b>Description</b>
joblib	Serialization and model saving utility.
protobuf	Efficient data serialization format by Google.
deep-translator	Multilingual text translation.
Requests	Library for making HTTP requests.
google-generativeai	Integrate Google's generative AI models.
ffmpeg	Multimedia framework for processing audio/video.
tesseract-ocr	OCR tool for text extraction.
portaudio19-dev	Required for handling audio input/output.
poppler-utils	Converts PDFs to images for text extraction.
pydub	Audio processing library.
sounddevice	Real-time audio recording/playback.
wavio	WAV file handling.
numpy	Numerical computing library.
PyAudio	Audio input/output handling.
SpeechRecognition	Converts speech to text.
pdf2image	Converts PDFs to images.
tweepy	Access and analyze Twitter data.
openai-whisper	Speech-to-text model from OpenAI.
tiktoken	Tokenizer for language models.
librosa	Audio analysis and feature extraction.
opencv-python	Computer vision and image processing.
xgboost	Gradient boosting for ML models.
deepface	Facial recognition and emotion analysis.
tf_keras	Keras API for TensorFlow.
transformers	Pre-trained NLP models from Hugging Face.
tensorflow	Deep learning framework.
nltk	Natural language processing toolkit.
plotly	Interactive data visualization.
matplotlib	Static and animated plots.
scipy	Scientific computing and signal processing.
networkx	Graph analysis and visualization.
yt-dlp	YouTube video/audio downloader.

**Libraries and Packages Used**

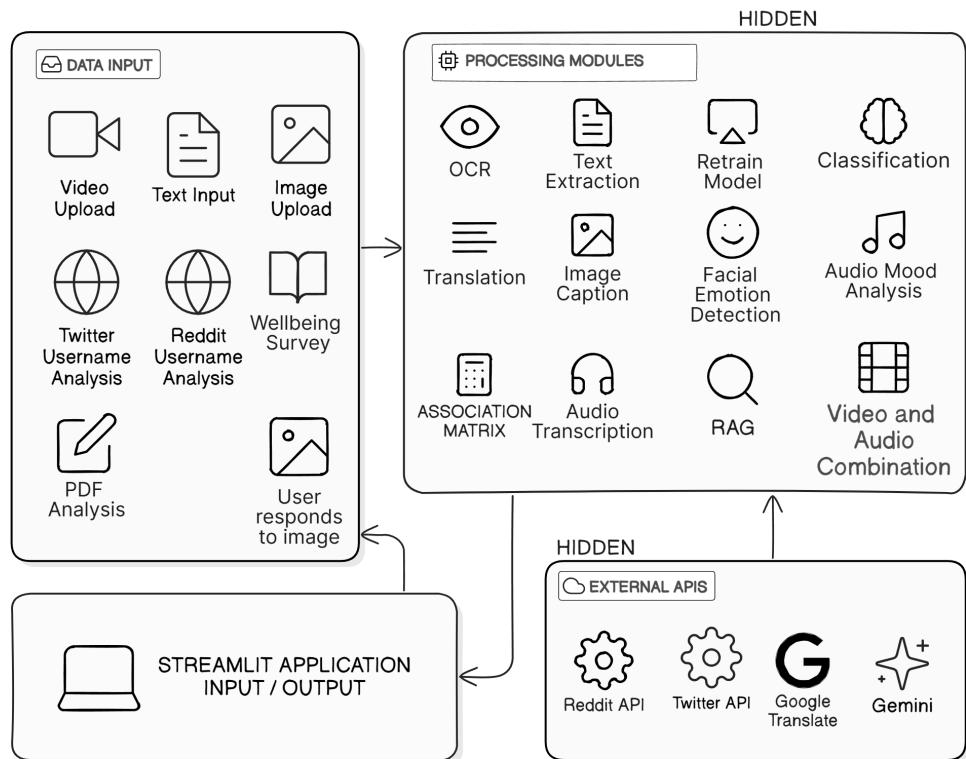
## 7.2 Hierarchy of Modules



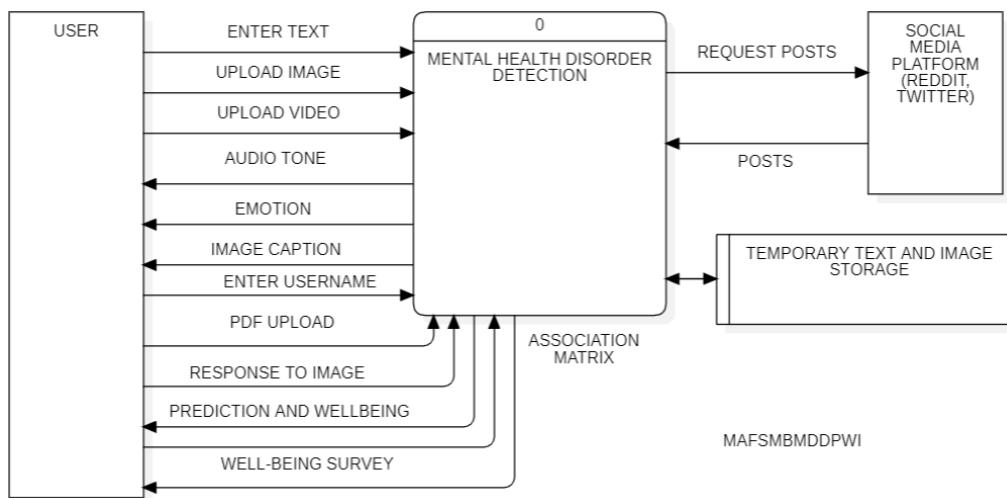
Module	Description
<b>Data Collection</b>	Gathers relevant text data from posts in platforms like Reddit via PRAW.
<b>Data Preprocessing</b>	<p>Cleans and prepares text data using:</p> <ul style="list-style-type: none"> <li>Tokenization (splitting text into words/tokens)</li> <li>Lowercasing for uniformity</li> <li>Stop-word removal</li> <li>Lemmatization or stemming</li> </ul> <p>Perform feature extraction to convert text into numerical features using:</p> <ul style="list-style-type: none"> <li>Term Frequency-Inverse Document Frequency (TF-IDF)</li> <li>Bag of Words (BoW) model</li> <li>Word2Vec, LIWC (Linguistic Inquiry and Word Count) and N-Gram were also explored</li> </ul>
<b>Model Training and Validation</b>	<p>Splits dataset into training/testing sets and trains models such as:</p> <ul style="list-style-type: none"> <li>Logistic Regression, Naive Bayes, SVM, XGBoost, KNN, LSTM, Transformer. These formed the base models for the ensemble model used in application where Random Forest was used as the meta-learner. All were validated to assess performance.</li> </ul>
<b>Prediction</b>	Processes received and combined text input for classification using trained models.
<b>Testing and Deployment</b>	Deploys models on Streamlit Cloud for real-time predictions and wellbeing insights using association matrix. Provides a user interface for easy access.

**Hierarchy of Modules in the system**

### 7.3 Detailed Design

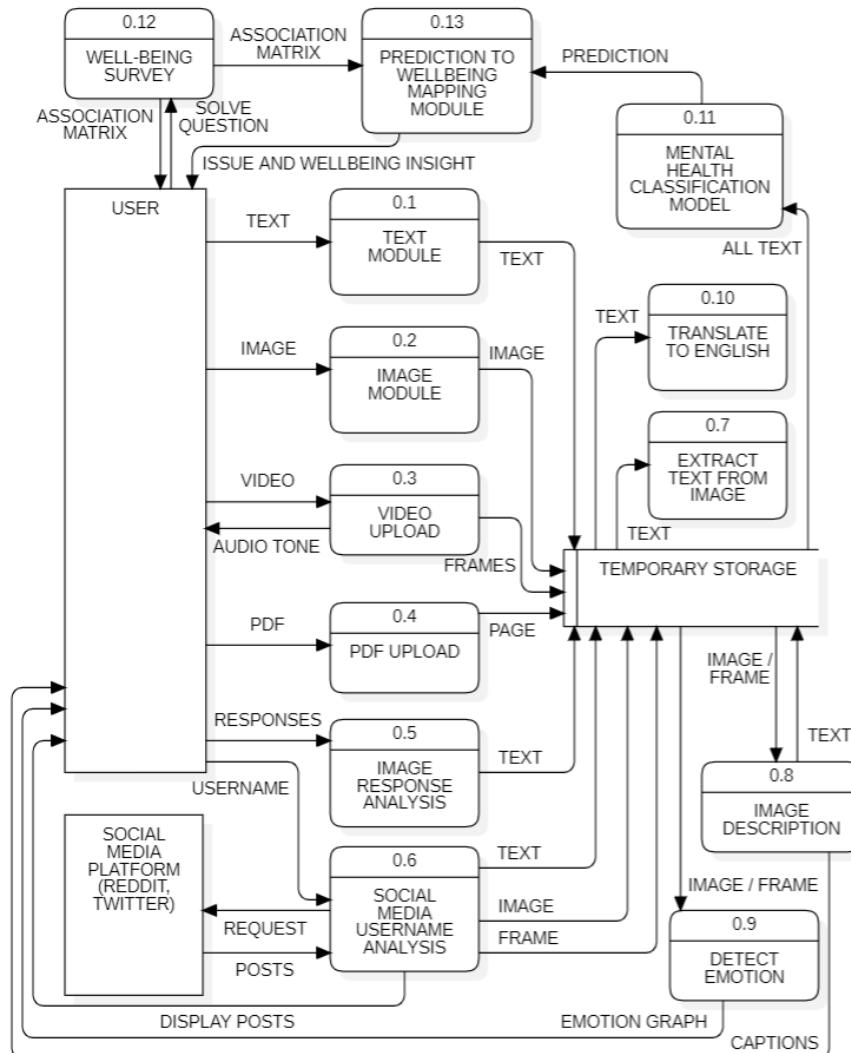


System Overview



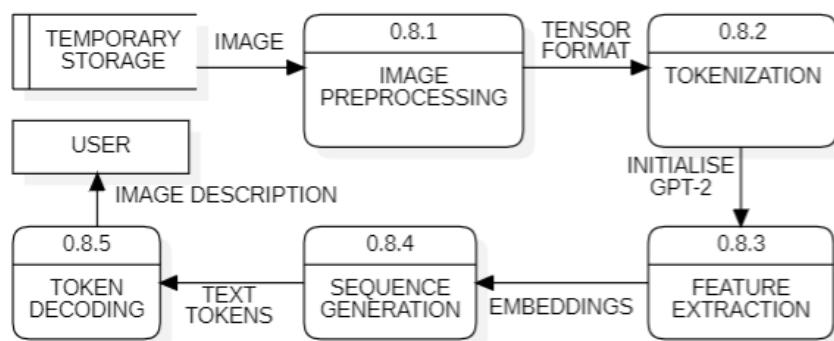
DFD Level 0 of the System

# MAFSMBMDDPW



DFD Level 1 of the System

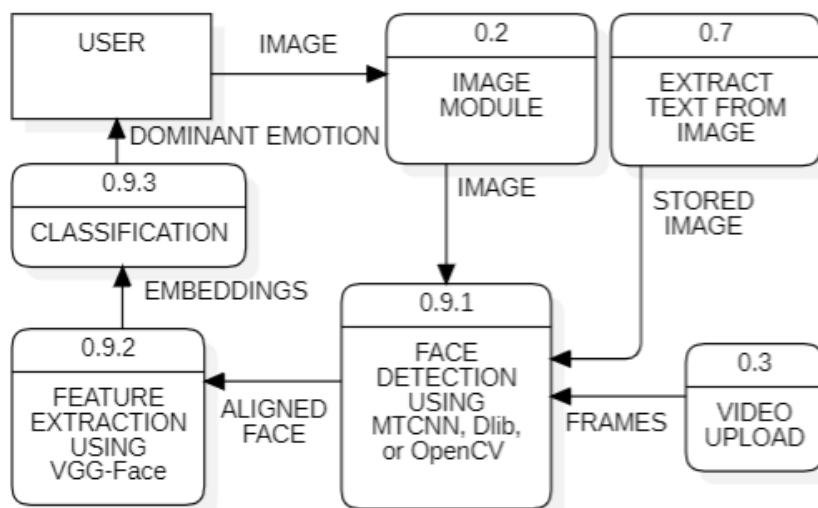
## Image Description



DFD Level 2 of Image Description

The image captioning process using Python's Transformers module and the ViT-GPT2 model involves several key steps. The user uploads an image, which is preprocessed by resizing to  $224 \times 224$  pixels, normalizing pixel values, and converting it into a tensor format. The Vision Transformer (ViT) divides the image into  $16 \times 16$  patches, embeds them, and processes them through transformer layers to extract visual features. The resulting embedding is passed to GPT-2, which generates a sequence of text tokens iteratively using its language model. Tokens are decoded back into human-readable text using the GPT-2 tokenizer, cleaned for readability, and output as a concise description. This pipeline leverages ViT for feature extraction and GPT-2 for language generation, enabling efficient and almost accurate image captioning.

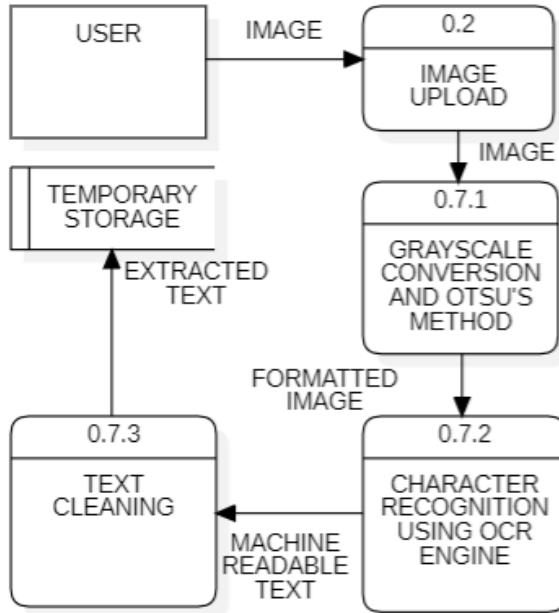
### Emotion Detection Functionality



DFD Level 2 of Emotion Detection Functionality

DeepFace analysis processes facial features from uploaded images or video frames through a structured pipeline. It starts with face detection using models like MTCNN, Dlib, or OpenCV to locate and align faces. The detected faces are cropped and passed to feature extraction, where pre-trained models like VGG-Face or Facenet generate numerical embeddings. These embeddings are compared using cosine similarity or Euclidean distance for tasks like emotion detection (e.g., happiness, sadness), demographic analysis (e.g., age, gender), or face verification. The results, such as detected emotions or attributes, are then displayed to the user. This process enables real-time analysis of facial expressions and emotions, providing valuable insights for mental health monitoring and wellbeing assessment.

## Extract Text From Image



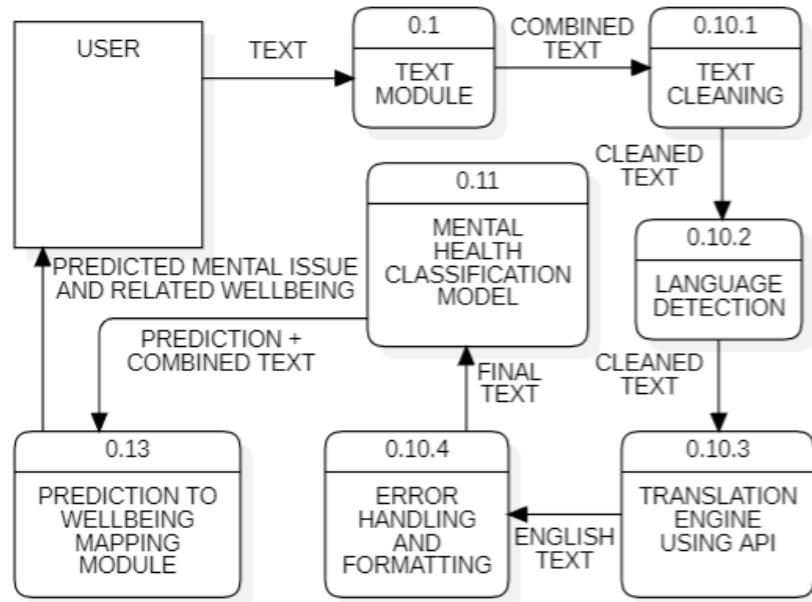
DFD Level 2 of Text Extraction from Image

The process of extracting text from an image using Tesseract-OCR begins with preprocessing the uploaded image. The image is converted to grayscale, noise is reduced using Gaussian or Median Blur, and binarization (e.g., Otsu's thresholding) isolates text from the background. Text regions are detected using contour analysis or connected components. The processed image is then passed to Tesseract, which uses an LSTM-based neural network to recognize characters and generate machine-readable text, enhanced by language models for accuracy. Postprocessing corrects errors via spell-checking and rule-based replacements (e.g., 0 → O), and formats the text into paragraphs or lines. The final output, displayed or saved as .txt or .docx, is ready for applications like document analysis. This pipeline combines image enhancement, deep learning, and text correction for high-quality results.

## Translation to English

The process of translating text to English using DeepTranslator begins with preprocessing the input to clean unwanted characters and detect the source language using machine learning models like Google's Compact Language Detector. The prepared text is passed to DeepTranslator, which interfaces with APIs like Google Translate or Microsoft Translator. These APIs use neural machine translation (NMT) with encoder-decoder architectures and attention mechanisms to translate the text into English. Transformer-based models enhance accuracy by capturing context and long-range dependencies. Postprocessing ensures quality by validating completeness, correcting errors, and preserving formatting. The final translated text is displayed or saved,

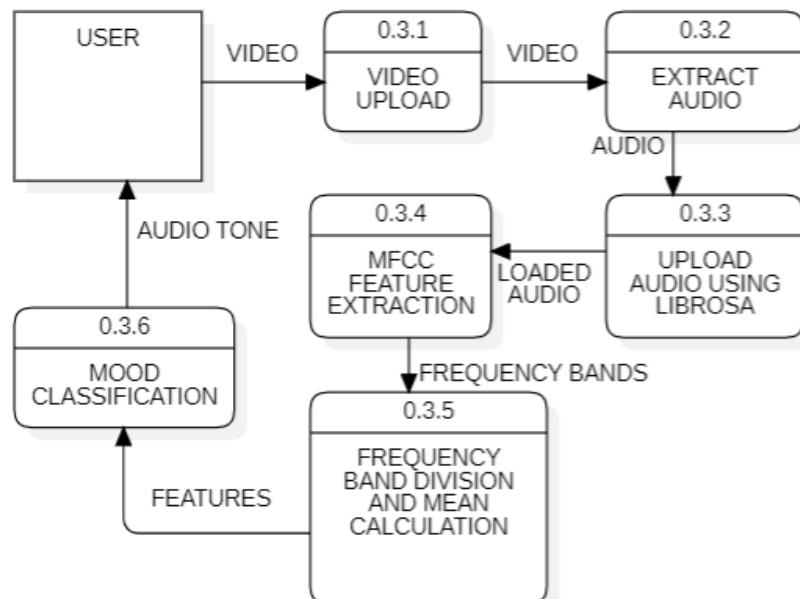
## MAFSMBMDDPW



DFD Level 2 of Translation to English

ensuring accuracy and readability. This pipeline integrates NLP, deep learning, and error handling for reliable translations.

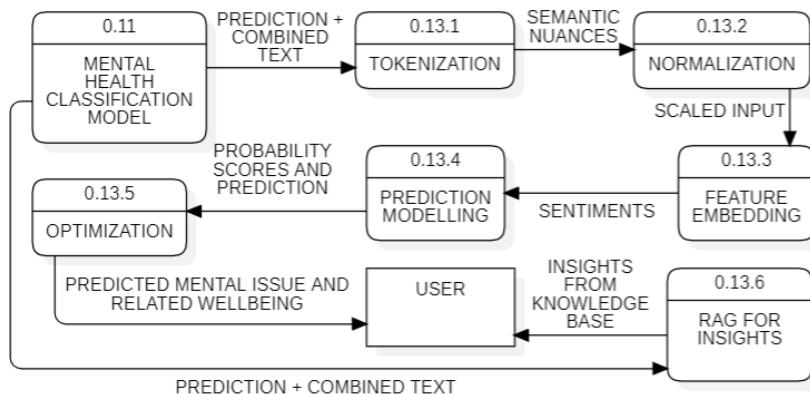
### Audio Mood Analysis



DFD Level 2 of Audio Mood Analysis

The `analyze_audio_mood` function begins with the user providing a video file path. The audio is extracted using the `extract_audio_from_video` function and loaded into memory with the Librosa library. Mel-frequency cepstral coefficients (MFCCs) are computed using `librosa.feature.mfcc` to capture frequency patterns for mood analysis. The MFCC array is segmented into four frequency bands—low, mid-low, mid-high, and high—and the scalar mean of each band is calculated to simplify data for classification. The mood is classified (e.g., normal, calm, anxious) based on the dominant frequency characteristics. Results can be further enhanced using the Gemini API to provide tone, mood, and summary details for the audio. This process combines audio feature extraction and classification for comprehensive mood analysis.

## Prediction to Wellbeing Mapping



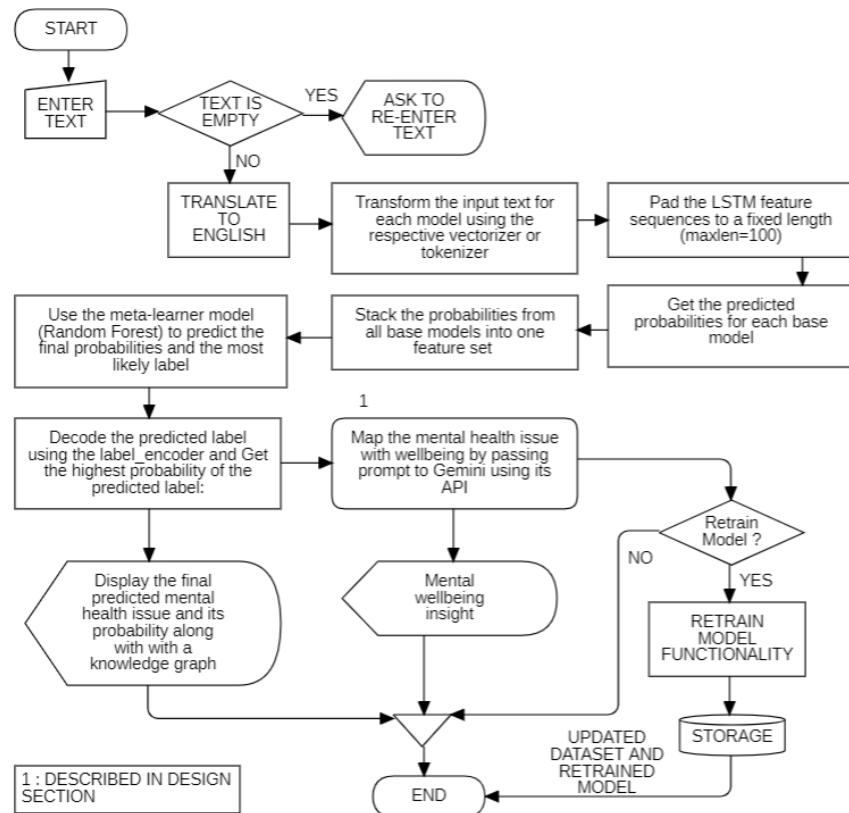
DFD Level 2 of Prediction to Wellbeing Mapping

The system first processes user input (text, behavioral data, or health metrics) through preprocessing and feature extraction. Machine learning models then predict the most likely mental health condition along with associated probabilities. The top predicted issue and its probability are then fed into GEMINI 2.0 FLASH with a structured prompt to generate wellbeing insights based on Ryff's six parameters. To refine these insights, an association matrix maps the probabilities of all predicted issues to specific wellbeing parameters, selecting 1 to 3 key parameters (e.g., autonomy, personal growth, or self-acceptance). This ensures that the user receives targeted recommendations for improving their psychological wellbeing. There is also an additional feature where Retrieval Augmented Generation (RAG) is used to provide personalized recommendations based on the user's mental health condition. The system retrieves relevant information from a knowledge base and generates tailored suggestions.

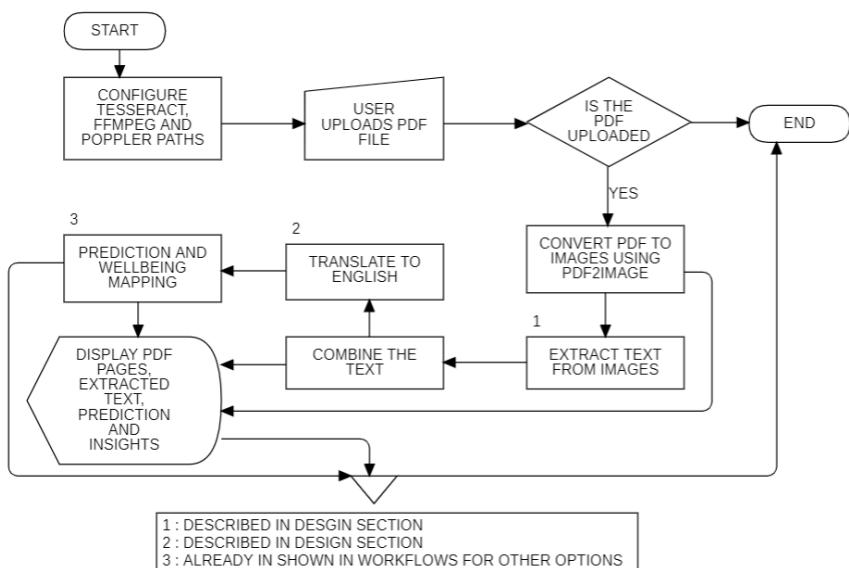
The above 6 main functionalities are reused in the application for the options available to the user. These include **Text** analysis, **Image** analysis, **Video** analysis, **PDF** analysis, analysis of **User**

**response to Image**, analysis of **Reddit** and **Twitter** user profiles. **Wellbeing Survey Option** and **RAG for wellbeing Insights** have been added under *Implementation* section.

Below are the flow diagrams for the various *Analysis* options that the web application provides.



Text Classification Flow Diagram



PDF Upload Flow Diagram

## MAFSMBMDDPW

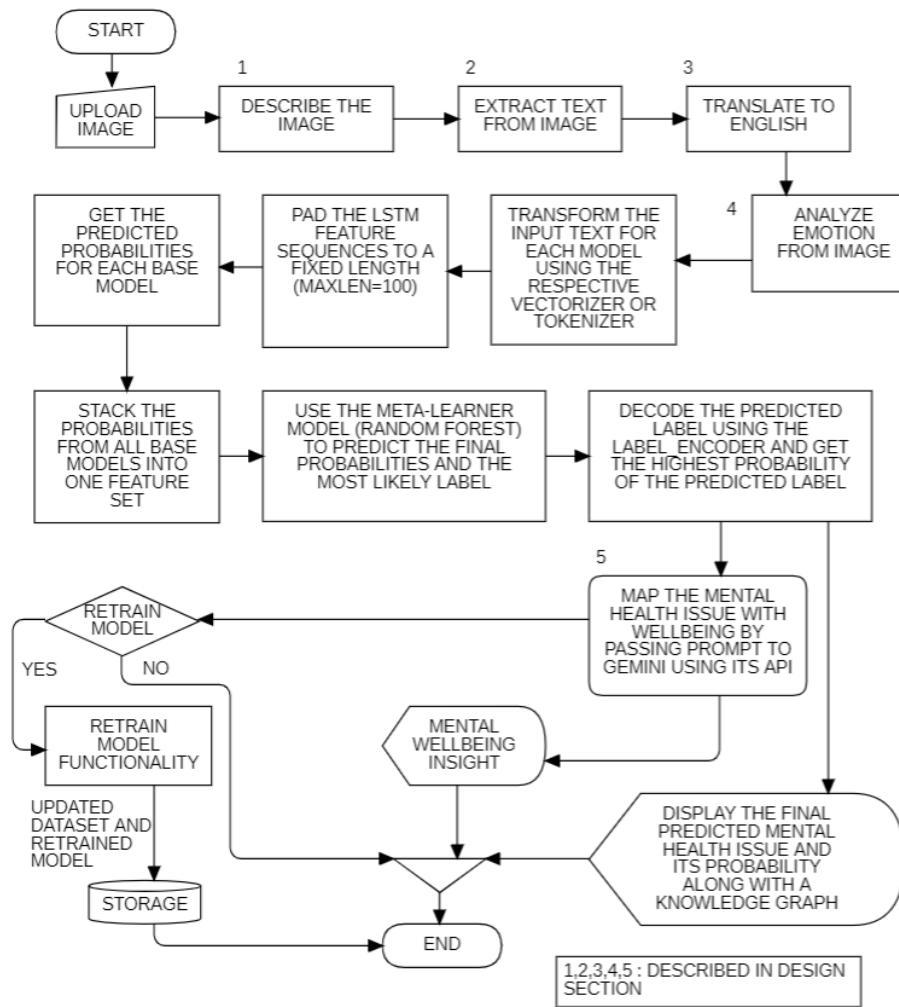
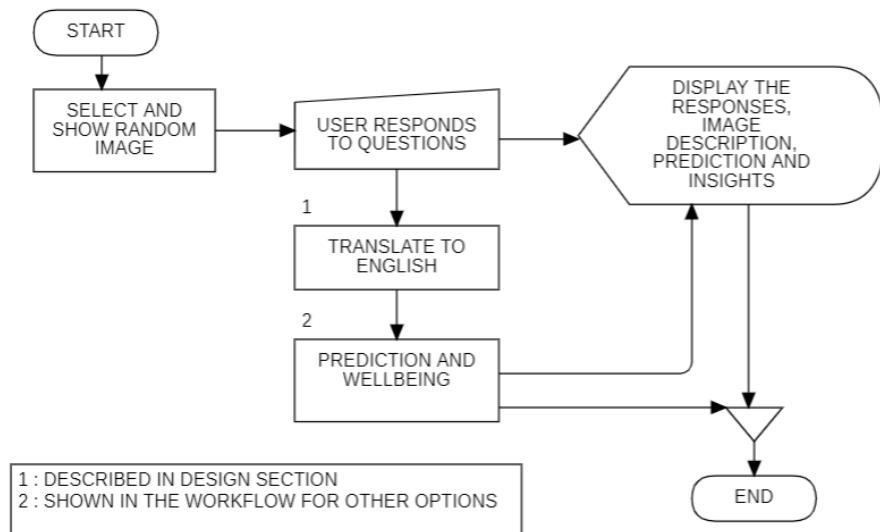
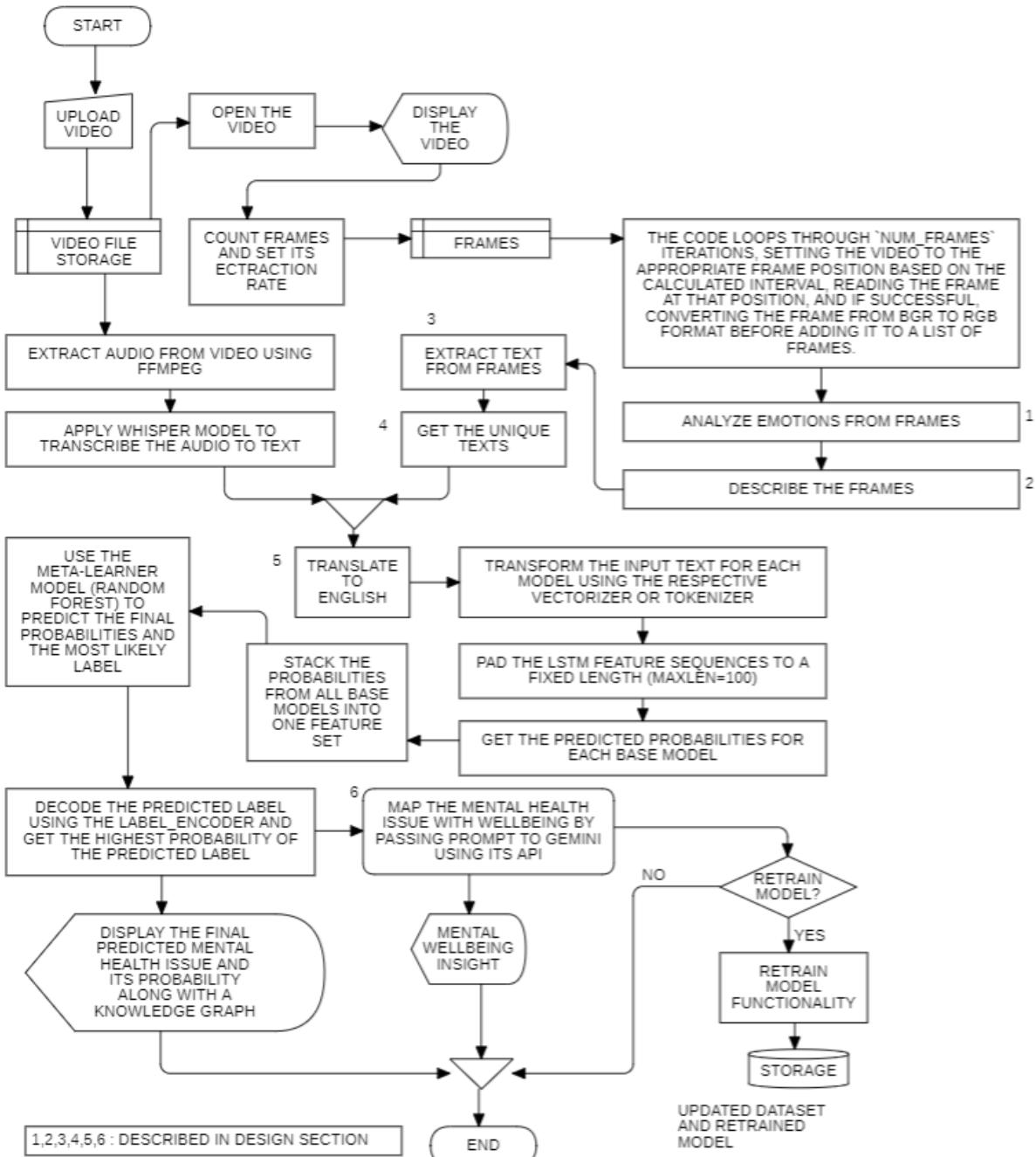


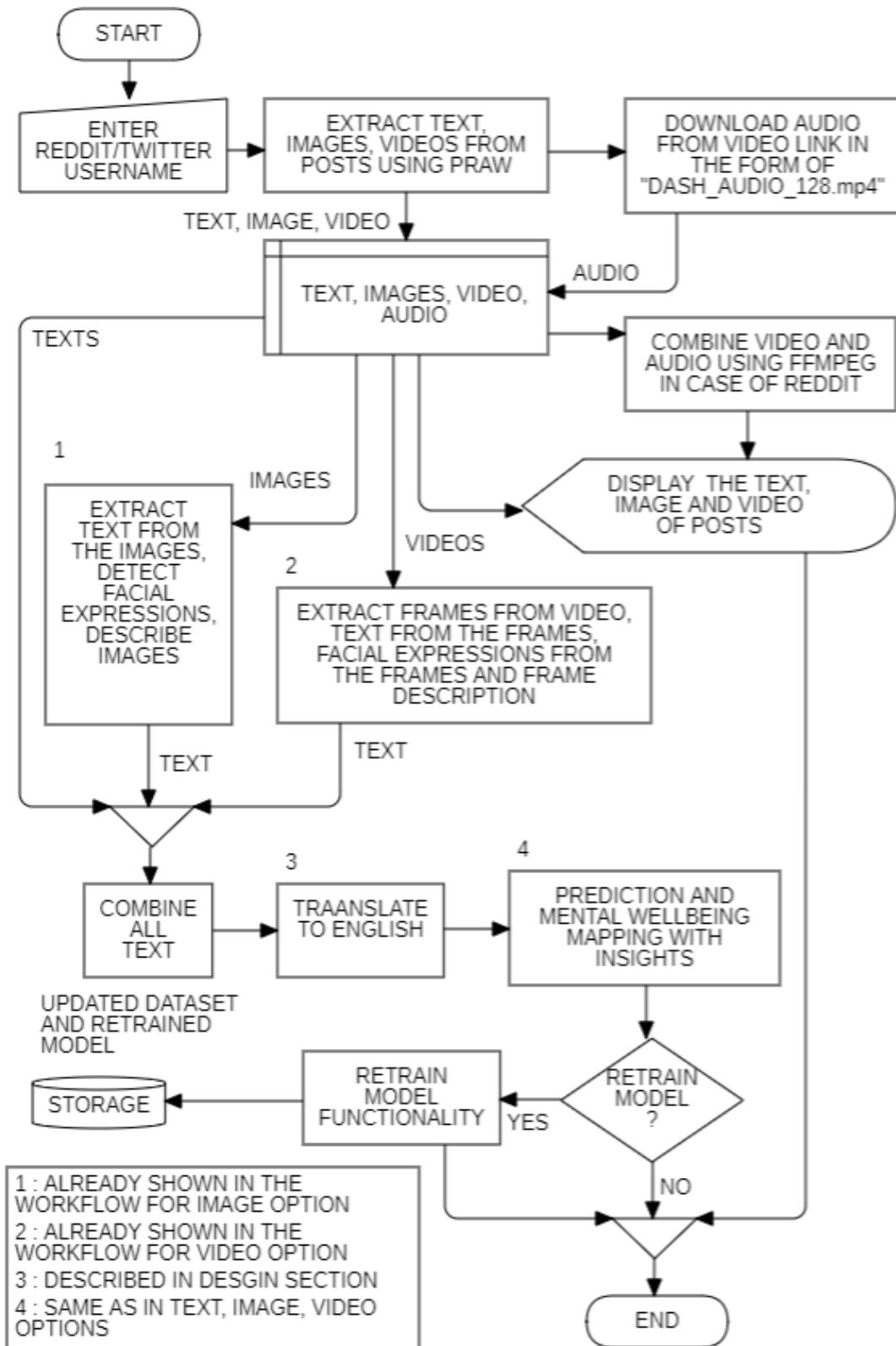
Image Classification Flow Diagram



User Response to Image Flow Diagram

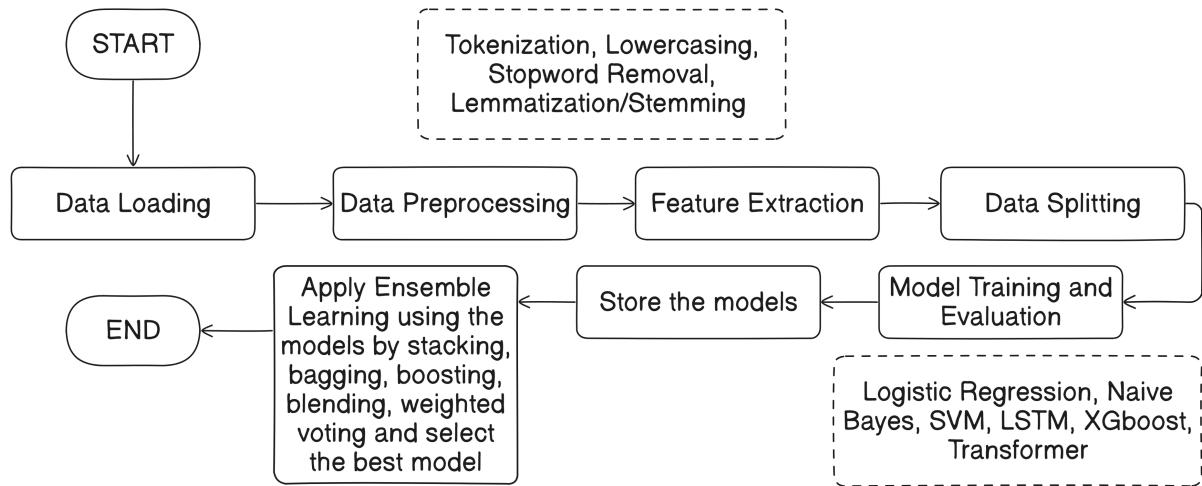


Video Classification Flow Diagram



Reddit and Twitter username Classification Flow Diagram

## 8 Implementation



Workflow for getting the model for the web application

### 8.1 Data Collection and Dataset Preparation

Stepwise Algorithm for Data Collection and Dataset Combination

Step	Description
1	<b>Import Libraries:</b> Import praw, pandas, time for data collection and sklearn.utils.shuffle for combining datasets.
2	<b>Initialize Reddit API:</b> Use the provided credentials (client_id, client_secret, user_agent) to create a Reddit instance.
3	<b>Define Subreddits and Labels:</b> Create a dictionary mapping labels to subreddit lists: <ul style="list-style-type: none"> <li><b>normal:</b> news, AskReddit</li> <li><b>depression:</b> depression</li> <li><b>ptsd:</b> PTSD</li> <li><b>anxiety:</b> Anxiety</li> <li><b>bipolar:</b> BipolarReddit</li> </ul>
4	<b>Set Post Types and Limit:</b> Define post types (hot, new, top) and set posts_per_type to 100.
5	<b>Collect Posts:</b> <ul style="list-style-type: none"> <li>For each label and its associated subreddits, iterate over each post type.</li> <li>Retrieve posts from the subreddit (using the corresponding post type and a limit of 100).</li> <li>For each post, combine the title and selftext, and append the result along with its label to a data list.</li> <li>Pause for 1 second between requests.</li> </ul>

## Stepwise Algorithm for Data Collection and Dataset Combination

Step	Description
6	<b>Save Collected Data:</b> Convert the data list into a DataFrame with columns <code>text</code> and <code>label</code> and save it as <code>{label}_dataset.csv</code> .
7	<b>Load Datasets:</b> Read the individual CSV files for <code>bipolar</code> , <code>depression</code> , <code>normal</code> , <code>anxiety</code> , and <code>ptsd</code> into separate DataFrames.
8	<b>Determine Minimum Length:</b> Compute <code>min_length</code> as the minimum number of records among datasets (using <code>len(normal_df) // 6</code> for the normal dataset to balance its count).
9	<b>Create a Balanced Pattern:</b> <ul style="list-style-type: none"> <li>Loop from 0 to <code>min_length - 1</code>.</li> <li>For each iteration, append to a new list:           <ul style="list-style-type: none"> <li>The <math>i^{\text{th}}</math> record from <code>bipolar_df</code>.</li> <li>The <math>i^{\text{th}}</math> record from <code>depression_df</code>.</li> <li>Six consecutive records from <code>normal_df</code> (indices <math>i * 6</math> to <math>(i+1) * 6</math>).</li> <li>The <math>i^{\text{th}}</math> record from <code>anxiety_df</code>.</li> <li>The <math>i^{\text{th}}</math> record from <code>ptsd_df</code>.</li> </ul> </li> </ul>
10	<b>Convert to DataFrame:</b> Transform the balanced list into a DataFrame ( <code>pattern_df</code> ).
11	<b>Prepare Remaining Data:</b> Concatenate the leftover records from each dataset (beyond <code>min_length</code> for all datasets and beyond <code>min_length * 6</code> for the normal dataset) and shuffle them.
12	<b>Merge and Save Final Dataset:</b> Combine <code>pattern_df</code> with the shuffled remaining data, reset the index, and save the final DataFrame as <code>mental_health_combined.csv</code> .

```

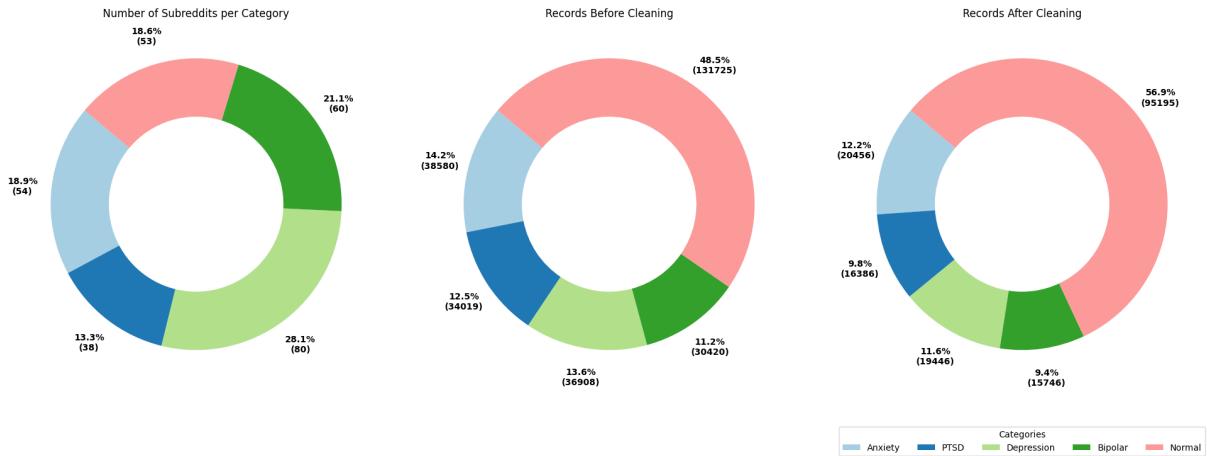
text mental_health_issue
0 2024 Election Due to the 2024 US Presidential ...      bipolar
1 Our most-broken and least-understood rules is ...      depression
2 Rules **UPDATED** 1. If you're here you must p...      normal
3 Happy Cakeday, r/Normal! Today you're 11 Let's...      normal
4 Happy Cakeday, r/Normal! Today you're 10 Let's...      normal
5 I am also a person I am very normal. Today I t...      normal
6 Happy Cakeday, r/Normal! Today you're 9 Let's ...      normal
7 I am a person I am a person                         normal
8 Elections and Politics Hello friends!\n\nIt's ...      anxiety
9 You are more than just one emotion                  ptsd

```

Obtained Dataset

The dataset for mental health classification was compiled from subreddit communities, initially containing 385,80 records across five categories: anxiety (54 subreddits), PTSD (38), depression (80), bipolar disorder (60), and normal mental states (53). After data cleaning to ensure quality and relevance, the dataset was reduced to 167,279 records. A subset of 18,596 cleaned records was used for analysis to address computational constraints, such as memory and processing

# MAFSMBMDDPW



Collected Data Statistics

power, which would make training on the full dataset infeasible on standard hardware. This approach balances efficiency and performance, enabling faster experimentation and iterative model improvement while maintaining a representative sample.

## 8.2 Data Cleaning and Feature Extraction

### Stepwise Algorithm for Data Cleaning and Feature Extraction

Step	Description
1	<b>Import Libraries &amp; Resources:</b> Import pandas, re, TfidfVectorizer from scikit-learn, and NLTK modules. Download the necessary NLTK resources (e.g., stopwords, punkt, and punkt.tab).
2	<b>Load Dataset:</b> Read the CSV file mental_health.csv into a pandas DataFrame.
3	<b>Handle Missing Values:</b> Drop any rows where the text field is missing.
4	<b>Remove Duplicates:</b> Eliminate duplicate rows based on the text column.
5	<b>Define Negative Words:</b> Create a set of negative words (e.g., not, no, nor, etc.) that will be retained during stopword removal.
6	<b>Define Cleaning Function:</b> Write the function clean_text(text) to preprocess text by: <ol style="list-style-type: none"> <li>Removing URLs using regex.</li> <li>Removing mentions (e.g., @username).</li> <li>Removing special characters, numbers, and punctuation.</li> <li>Converting text to lowercase.</li> <li>Tokenizing the text using NLTK's word_tokenize.</li> <li>Removing stopwords (while keeping negative words).</li> <li>Rejoining tokens into a cleaned string.</li> </ol>

## Stepwise Algorithm for Data Cleaning and Feature Extraction

Step	Description
7	<b>Apply Cleaning Function:</b> Execute <code>clean_text</code> on the <code>text</code> column and store the result in a new column called <code>cleaned_text</code> .
8	<b>Initialize TF-IDF Vectorizer:</b> Create a TF-IDF vectorizer instance with a maximum of 10,000 features. (Other methods include Bag-Of-Words, LIWC, N-Gram, Word2Vec)
9	<b>Fit &amp; Transform Data:</b> Apply the vectorizer to the <code>cleaned_text</code> to generate the TF-IDF feature matrix <code>X</code> .
10	<b>Optional - Convert to DataFrame:</b> Convert the TF-IDF sparse matrix <code>X</code> into a pandas DataFrame for easier inspection (e.g., print the first few rows).
11	<b>Optional - Save Preprocessed Data:</b> Save the final preprocessed DataFrame to <code>preprocessed_mental_health.csv</code> .

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa ab abandon abandoned \
0 0.0          0.0 0.0      0.0 0.0
1 0.0          0.0 0.0      0.0 0.0
2 0.0          0.0 0.0      0.0 0.0
3 0.0          0.0 0.0      0.0 0.0
4 0.0          0.0 0.0      0.0 0.0

abandoning abandonment abc abilities ability ... zemeckis zemlja \
0      0.0      0.0 0.0      0.0 0.00000 ...      0.0 0.0
1      0.0      0.0 0.0      0.0 0.03726 ...      0.0 0.0
2      0.0      0.0 0.0      0.0 0.00000 ...      0.0 0.0
3      0.0      0.0 0.0      0.0 0.00000 ...      0.0 0.0
4      0.0      0.0 0.0      0.0 0.00000 ...      0.0 0.0

zero zoloft zombie zombieland zombies zone zoom zuckerberg
0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

[5 rows x 10000 columns]
```

After TF-IDF Vectorization

**Datasets:****Before Cleaning:** `mental_health.csv`**After Cleaning:** `preprocessed_mental_health.csv`

Stage	Schema	Description
Before Cleaning	text: Original text data mental_issue: Mental health issues	Contains raw, unprocessed text data with possible missing values, duplicates, URLs, mentions, and special characters.
After Cleaning	text: Original text data mental_issue: Mental health issues cleaned_text: Processed text data	Data is cleaned by removing URLs, mentions, special characters, and extra noise. The text is converted to lowercase, tokenized, stopwords (except key negative words) are removed, and the cleaned text is stored in the new column cleaned_text.

Dataset Schema Before and After Data Cleaning

The matrix dimensions for Bag of Words are determined by the number of records, which is 18,597 in this case, and the size of the vocabulary, which represents the number of unique words in the cleaned\_text column. Similarly, for TF-IDF, the dimensions of the matrix are the same as BOW, calculated as the number of records multiplied by the vocabulary size. For N-gram, the matrix size depends on the range of n-grams used. For example, a unigram produces dimensions equivalent to BOW, while bigram or trigram models increase the vocabulary size due to the inclusion of word combinations. Word2Vec, on the other hand, creates a dense vector representation for each word, with dimensions based on the predefined vector size, such as 100 or 200. Aggregating these vectors at the sentence level, typically by averaging, results in a matrix of dimensions equal to the number of records multiplied by the vector dimensions. For LIWC, the dimensions are determined by the number of predefined LIWC categories, which is typically around 70. The resulting matrix dimensions are the number of records multiplied by the number of LIWC categories.

### 8.3 Machine Learning Models

Algorithm like Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, XGBoost and Naive Bayes are implemented for the classification of mental health issues. The code algorithm below demonstrates the implementation. The models are evaluated on the test set using metrics like accuracy and classification reports. Hyperparameter Tuning is performed using RandomizedSearchCV to optimize the model performance. Naive Bayes model after hyperparamter tuning is selected along with the basic Logistic Regression, Support Vector Machine, XGboost for the final ensemble model for the web application.

### Stepwise Algorithm for Machine Learning Models

Step	Description
1	<b>Data Loading and Verification:</b> Load the dataset ( <code>preprocessed_mental_health.csv</code> ) using pandas, verify required columns ( <code>cleaned_text</code> and <code>mental_health_issue</code> ), and drop rows with missing values.
2	<b>Tokenization and Feature Extraction:</b> Choose a method (Bag-of-Words, TF-IDF, LIWC, Word2Vec, or N-gram) and transform <code>cleaned_text</code> into a numerical feature matrix $X$ .
3	<b>Target Variable Preparation:</b> Extract the target variable $y$ from the <code>mental_health_issue</code> column.
4	<b>Dataset Splitting:</b> Split $X$ and $y$ into training and test sets (e.g., 80/20 split) using <code>train_test_split</code> with a fixed random state.
5	<b>Model Initialization:</b> For each algorithm (Logistic Regression, Naive Bayes, SVM, KNN, Random Forest, XGBoost), initialize the model with appropriate hyperparameters.
6	<b>Model Training:</b> Train the selected model on the training data.
7	<b>Prediction:</b> Use the trained model to predict mental health issues on the test set.
8	<b>Model Evaluation:</b> Evaluate performance by computing accuracy, classification reports, and confusion matrices.
9	<b>Cross-Validation:</b> Apply Stratified K-Fold (e.g., 5 folds) cross-validation to record accuracies and calculate the mean and standard deviation.
10	<b>Performance Comparison:</b> Compare evaluation metrics across all models and feature extraction methods to select the best combination.
11	<b>Hyperparameter Tuning:</b> For Logistic Regression, Naive Bayes, SVM, and KNN, optimize hyperparameters using <code>RandomizedSearchCV</code> to search the parameter space and select the optimal configuration.

## 8.4 Deep Learning Models

### Stepwise Algorithm for LSTM-based Model

Step	Description
1	<b>Data Loading:</b> Load <code>preprocessed_mental_health.csv</code> using pandas.
2	<b>Feature &amp; Target Preparation:</b> <ul style="list-style-type: none"> <li>Extract text (<math>X</math>) and target (<math>y</math>).</li> <li>Encode <math>y</math> using <code>LabelEncoder</code> and convert to one-hot with <code>to_categorical</code>.</li> </ul>
3	<b>Tokenization &amp; Padding:</b> <ul style="list-style-type: none"> <li>Initialize Keras Tokenizer with <code>num_words=25000</code>.</li> <li>Fit on <math>X</math>, convert texts to sequences, and pad to <code>max_length=128</code> using <code>pad_sequences</code>.</li> </ul>
4	<b>Cross-Validation Setup:</b> Define a 5-fold <code>StratifiedKFold</code> ( <code>shuffle=True, random_state=42</code> ).

## Stepwise Algorithm for LSTM-based Model

Step	Description
5	<b>LSTM Model Building:</b> For each fold, build a Keras Sequential model with: <ul style="list-style-type: none"> <li>• Embedding (vocab_size, 128, input_length=128)</li> <li>• LSTM(128, return_sequences=True)</li> <li>• Dropout (0.2)</li> <li>• LSTM(64)</li> <li>• Dropout (0.2)</li> <li>• Dense(64, activation='relu')</li> <li>• Dense(num_classes, activation='softmax')</li> </ul>
6	<b>Model Training:</b> Compile the model with optimizer='adamw' and loss='categorical_crossentropy'. Train for 20 epochs with a batch size of 16 using the training fold and validate on the validation fold.
7	<b>Evaluation per Fold:</b> <ul style="list-style-type: none"> <li>• Evaluate the model on the validation set to obtain loss and accuracy.</li> <li>• Predict probabilities on the validation set and compute the confusion matrix.</li> </ul>
8	<b>ROC Curve Calculation:</b> <ul style="list-style-type: none"> <li>• Concatenate true labels and predictions from all folds.</li> <li>• Binarize true labels and compute ROC curves and AUC for each class plus a micro-average.</li> </ul>
9	<b>Visualization:</b> Plot ROC curves, validation loss/accuracy across folds, and the average confusion matrix using Matplotlib and Seaborn.
10	<b>Final Evaluation:</b> Compute the average cross-validated accuracy and display a classification report.

```

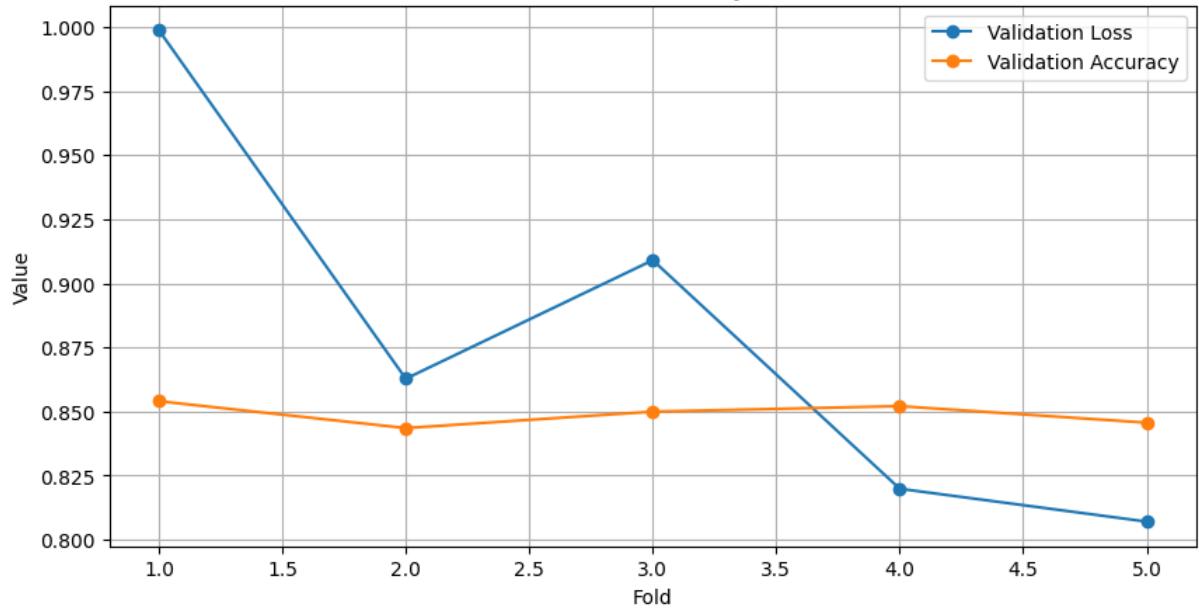
Epoch 1/10
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
warnings.warn(
465/465 135s 280ms/step - accuracy: 0.6046 - loss: 1.1142 - val_accuracy: 0.6841 - val_loss: 0.7390
Epoch 2/10
465/465 131s 281ms/step - accuracy: 0.6681 - loss: 0.7686 - val_accuracy: 0.6634 - val_loss: 0.8092
Epoch 3/10
465/465 144s 286ms/step - accuracy: 0.6499 - loss: 0.8130 - val_accuracy: 0.6680 - val_loss: 0.7538
Epoch 4/10
465/465 139s 280ms/step - accuracy: 0.6368 - loss: 0.8657 - val_accuracy: 0.6605 - val_loss: 0.7481
Epoch 5/10
465/465 130s 280ms/step - accuracy: 0.6949 - loss: 0.6574 - val_accuracy: 0.6922 - val_loss: 0.6711
Epoch 6/10
465/465 141s 278ms/step - accuracy: 0.7132 - loss: 0.6056 - val_accuracy: 0.6933 - val_loss: 0.6697
Epoch 7/10
465/465 147s 289ms/step - accuracy: 0.7304 - loss: 0.5676 - val_accuracy: 0.7081 - val_loss: 0.6606
Epoch 8/10
465/465 131s 282ms/step - accuracy: 0.7483 - loss: 0.5433 - val_accuracy: 0.7516 - val_loss: 0.6319
Epoch 9/10
465/465 142s 282ms/step - accuracy: 0.8079 - loss: 0.4523 - val_accuracy: 0.7484 - val_loss: 0.6037
Epoch 10/10
465/465 142s 282ms/step - accuracy: 0.8564 - loss: 0.3823 - val_accuracy: 0.8355 - val_loss: 0.5368

```

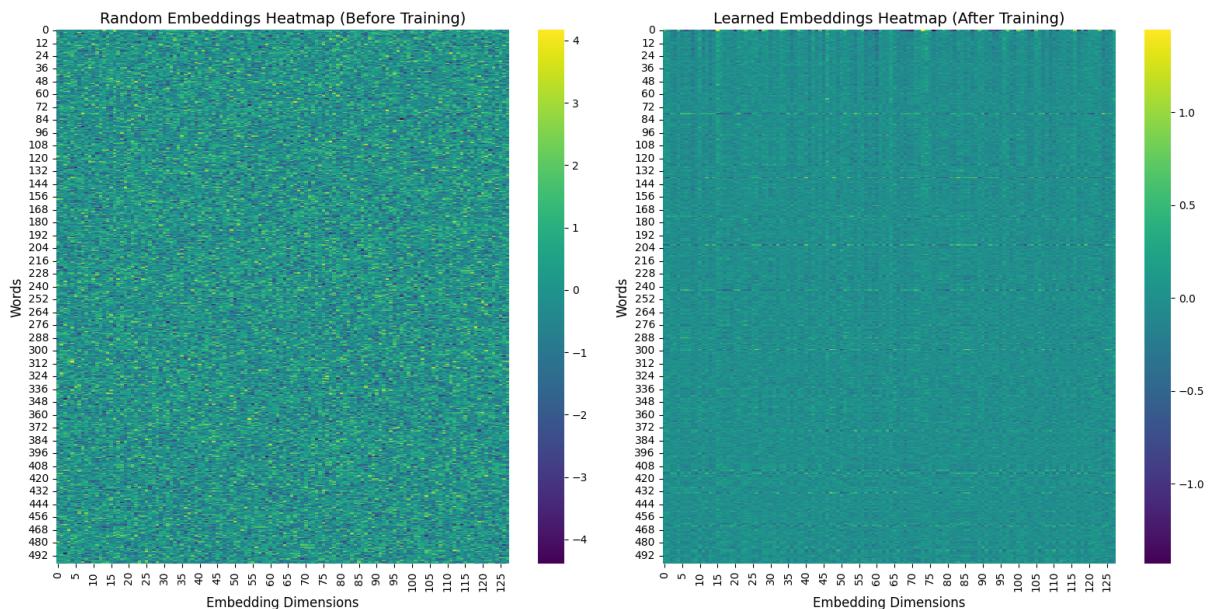
Output for LSTM Epochs

# MAFSMBMDDPW

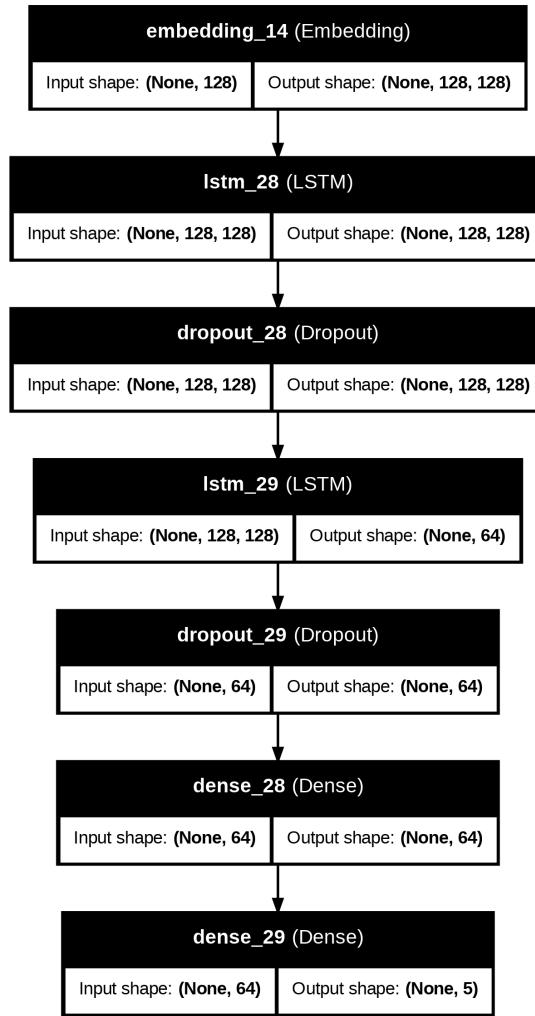
Validation Loss and Accuracy Across Folds



LSTM Validation loss and accuracy



LSTM Random and Learned Embeddings



LSTM Model Architecture

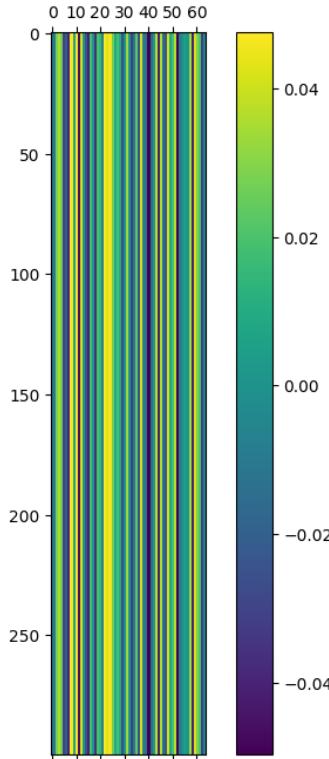
### Stepwise Algorithm for Custom Transformer-based Model

Step	Description
1	<b>Import Libraries:</b> Import required packages including NumPy, pandas, Matplotlib, Seaborn, scikit-learn modules (e.g., LabelEncoder, confusion_matrix), and TensorFlow Keras modules (e.g., TextVectorization, Embedding, LSTM, MultiHeadAttention, etc.).
2	<b>Load Dataset:</b> Use Google Colab's file upload to import preprocessed_mental_health.csv. Drop rows with missing cleaned_text and extract features (texts) and labels (labels).
3	<b>Preprocess Labels:</b> Encode labels with LabelEncoder, convert them to one-hot format via to_categorical, and retrieve class names.
4	<b>Text Vectorization:</b> Set vocab_size = 25000 and sequence_length = 300. Create a TextVectorization layer, adapt it on the input texts (using a TensorFlow Dataset), and vectorize the texts.

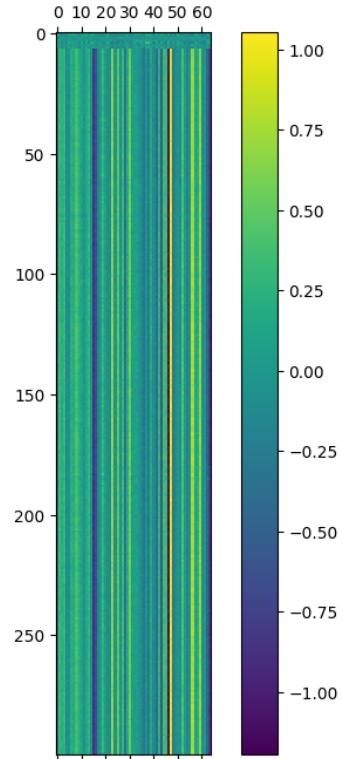
### Stepwise Algorithm for Custom Transformer-based Model

Step	Description
5	<p><b>Define Custom Transformer Model:</b></p> <ul style="list-style-type: none"> <li>• <b>EmbeddingLayer:</b> Custom layer that sums word embeddings with positional embeddings.</li> <li>• <b>EncoderLayer:</b> Custom layer using MultiHeadAttention, followed by a two-layer Dense network (with ReLU activation) and LayerNormalization.</li> <li>• <b>Model Architecture:</b> Build the model with an Input layer → EmbeddingLayer → EncoderLayer → GlobalAveragePooling1D → Dense (ReLU) → Output Dense (softmax).</li> <li>• Compile with optimizer <code>adamw</code> and loss <code>categorical_crossentropy</code>.</li> </ul>
6	<p><b>Train the Model:</b> Convert vectorized texts to a NumPy array and split into training and validation sets (80/20 split, <code>random_state=42</code>). Train the model for 5 epochs with a batch size of 32.</p>
7	<p><b>Evaluate the Model:</b></p> <ul style="list-style-type: none"> <li>• Evaluate validation loss and accuracy.</li> <li>• Predict on the validation set and convert predictions to class labels.</li> <li>• Compute the confusion matrix using scikit-learn's <code>confusion_matrix</code>.</li> </ul>
8	<p><b>Visualization:</b> Plot the confusion matrix with <code>ConfusionMatrixDisplay</code> (using Matplotlib) and optionally visualize random and learned embeddings.</p>

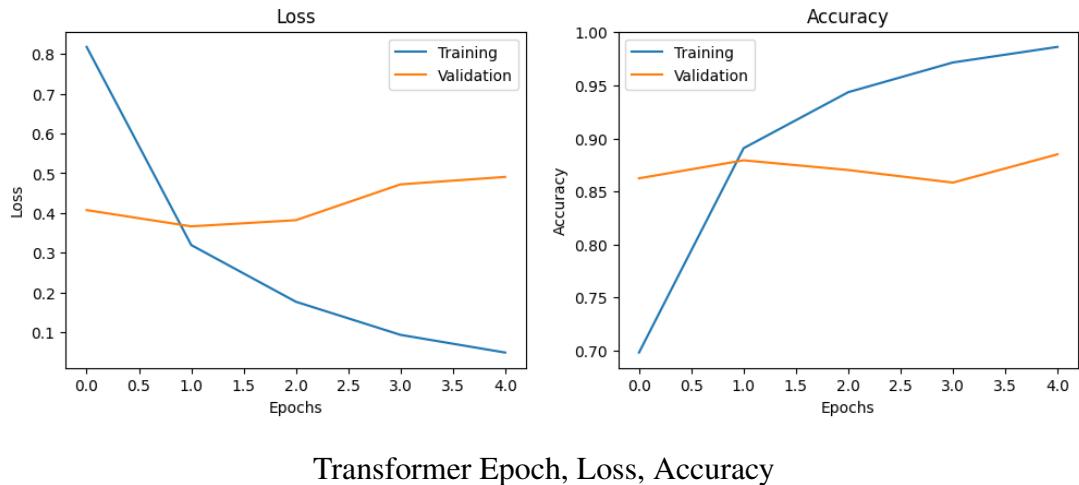
Random Embeddings



Learned Embeddings



Transformer Model Random and Learned Embeddings



## 8.5 Ensemble Model

### General Stepwise Algorithm for Ensemble Models

Step	Description
1	<b>Import Libraries:</b> Import required packages including numpy, pandas, tensorflow (Keras modules such as MultiHeadAttention, Embedding, etc.), scikit-learn (e.g., RandomForestClassifier, train_test_split, confusion_matrix), and pickle.
2	<b>Load Pre-trained Base Models &amp; Vectorizers:</b> Load saved models and vectorizers using pickle and load_model. Base models include: Logistic Regression (LR), Support Vector Machine (SVM), Naive Bayes (NB), XGBoost, LSTM, and Transformer. Also load corresponding vectorizers/tokenizers (e.g., LRvectorizer, SVMvectorizer, NBvectorizer, tfidf_vectorizer, LSTM_tokenizer, Tvectorize_layer) and label encoders.
3	<b>Load &amp; Preprocess Test Data:</b> Read preprocessed_mental_health.csv with pandas, drop rows with missing cleaned_text, extract texts and labels, and encode labels using LabelEncoder.
4	<b>Text Preprocessing for Each Model:</b> Transform the raw texts using respective vectorizers/tokenizers: <ul style="list-style-type: none"> <li>• LR, SVM, NB: Use LRvectorizer, SVMvectorizer, NBvectorizer.</li> <li>• XGBoost: Use tfidf_vectorizer.</li> <li>• LSTM: Convert texts using LSTM_tokenizer and apply pad_sequences.</li> <li>• Transformer: Process texts using Tvectorize_layer.</li> </ul>
5	<b>Generate Base Model Predictions:</b> For each base model, compute prediction probabilities: <ul style="list-style-type: none"> <li>• LR: lr_model.predict_proba</li> <li>• SVM: svm_model.predict_proba</li> <li>• NB: nb_model.predict_proba</li> <li>• XGBoost: xgb_model.predict_proba</li> <li>• LSTM: lstm_model.predict</li> <li>• Transformer: transformer_model.predict</li> </ul>

### General Stepwise Algorithm for Ensemble Models

Step	Description
6	<b>Stack Predictions:</b> Horizontally stack all base model prediction probabilities to form a new feature matrix ( <code>stacked_features</code> ) for the meta-learner.
7	<b>Ensemble Configuration &amp; Data Splitting:</b> Split the stacked features and true labels (using <code>train_test_split</code> ) into training and testing sets. Specify ensemble configurations: <ul style="list-style-type: none"> <li>• <b>Ensemble Model 1:</b> Base models: LR, XGBoost; Meta-learner: XGBoost.</li> <li>• <b>Ensemble Model 2:</b> Base models: LR, NB, SVM, XGBoost, LSTM; Meta-learner: XGBoost.</li> <li>• <b>Ensemble Model 3:</b> Base models: LR, NB, SVM, XGBoost, LSTM; Meta-learner: Random Forest.</li> <li>• <b>Ensemble Model 4:</b> Same base models using Bagging.</li> <li>• <b>Ensemble Model 5:</b> Same base models using Blending.</li> <li>• <b>Ensemble Model 6:</b> Same base models using Weighted Voting.</li> <li>• <b>Ensemble Model 7:</b> Base models: LR, SVM, NB, LSTM, XGBoost, Transformer; Meta-learner: Random Forest.</li> </ul>
8	<b>Train Meta-Learner:</b> Train the meta-learner (e.g., Random Forest, XGBoost, or ensemble strategies like Bagging, Blending, Voting) on the training portion of the stacked feature matrix.
9	<b>Evaluate Ensemble Model:</b> Predict on the test set using the meta-learner and compute evaluation metrics: accuracy, classification report, and confusion matrix. Also, perform cross-validation (e.g., using <code>cross_val_score</code> ) to assess stability.
10	<b>Output Results:</b> Print final evaluation metrics (accuracy, report, confusion matrix) and cross-validation results.

## 8.6 Wellbeing Survey and Association Matrix

1	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	issue	p1	p2	p3	p4	p5	p6	Date
2	6	1	6	1	6	3	6	1	6	1	6	1	Normal	12	12	10	12	12	12	2025-02-07
3	1	6	2	5	2	5	1	6	1	6	1	6	Depression	2	4	4	2	2	2	2025-02-07
4	4	5	2	3	1	4	3	6	5	4	2	3	Anxiety	6	6	4	4	8	6	2025-02-07
5	1	4	6	5	1	2	5	4	4	5	3	4	Bipolar	4	8	6	8	6	6	2025-02-07
6	1	4	3	4	1	6	5	4	3	6	1	4	PTSD	4	6	2	8	4	4	2025-02-07
7	3	4	1	2	1	4	1	4	6	5	2	3	Anxiety	6	6	4	4	8	6	2025-02-07
8	2	5	3	2	3	4	3	4	3	2	3	4	Bipolar	4	8	6	6	8	6	2025-02-07
9	3	4	1	2	1	4	1	4	6	5	2	3	Anxiety	6	6	4	4	8	6	2025-02-07
10	5	1	5	1	5	3	5	1	5	2	5	1	Normal	11	11	9	11	10	11	2025-02-07
11	2	5	4	5	1	6	5	2	5	3	6	3	PTSD	4	6	2	10	9	10	2025-02-07

Sample Response Collection Sheet

## Step-by-Step Algorithm for the Well-being Survey Option

Step	Operation	Description / Formula
1	<b>Initialization:</b> - Define file paths: csv_file_path, image_path, am_file_path. - Display title "Well-being Survey".	-
2	<b>Display Introductory Image:</b> - Check if the image file exists at image_path and display it.	-
3	<b>Display Instructions:</b> - Present informational messages and detailed instructions. - Describe the Ryff Scale	<p style="text-align: center;"> <b>1</b> → Strongly Disagree  <b>2</b> → Disagree  <b>3</b> → Slightly Disagree  <b>4</b> → Slightly Agree  <b>5</b> → Agree  <b>6</b> → Strongly Agree       </p>
4	<b>Load or Create Response Data:</b> - Attempt to load response.csv into a DataFrame. - If not found, create a new DataFrame with columns: {Q1, Q2, ..., Q12, issue, p1, ..., p6, Date}. - (R) against questions {Q2, Q4, Q6, Q8, Q10, Q12} signify reverse scoring	<p style="text-align: center;"> <b>Q1-Q2</b> : Self Acceptance  <b>Q3-Q4</b> : Positive relations with others  <b>Q5-Q6</b> : Autonomy  <b>Q7-Q8</b> : Environmental Mastery  <b>Q9-Q10</b> : Purpose In Life  <b>Q11-Q12</b> : Personal Growth       </p>
5	<b>Collect User Responses:</b> - Initialize an empty dictionary responses. - <b>Q00:</b> Record predicted mental issue (via radio buttons with options: Anxiety, Bipolar, Depression, Normal, PTSD). - <b>Q01-Q12:</b> Present 12 survey questions; record responses (scale 1–6).	<b>Follow these links for details:</b>  <ul style="list-style-type: none"> <li>- <a href="https://positivepsychology.com/ryff-scale-psychological-wellbeing/">https://positivepsychology.com/ryff-scale-psychological-wellbeing/</a></li> <li>- <a href="https://centerofinquiry.org/uncategorized/ryff-scales-of-psychological-well-being/">https://centerofinquiry.org/uncategorized/ryff-scales-of-psychological-well-being/</a></li> </ul>
6	<b>Append Current Date:</b> - Add the current date in YYYY-MM-DD format to responses.	Date = current date (YYYY-MM-DD)
7	<b>Overall Score Calculation:</b> For each well-being parameter, compute a score using two survey responses. <b>Variables:</b> S <sub>SA</sub> : Self Acceptance S <sub>PR</sub> : Positive Relations with Others S <sub>A</sub> : Autonomy S <sub>EM</sub> : Environmental Mastery S <sub>PL</sub> : Purpose in Life S <sub>PG</sub> : Personal Growth	$S_{SA} = Q1 +  7 - Q2 ,$ $S_{PR} = Q3 +  7 - Q4 ,$ $S_A = Q5 +  7 - Q6 ,$ $S_{EM} = Q7 +  7 - Q8 ,$ $S_{PL} = Q9 +  7 - Q10 ,$ $S_{PG} = Q11 +  7 - Q12 $

## Step-by-Step Algorithm for the Well-being Survey Option

Step	Operation	Description / Formula
8	<b>Determine Score Levels:</b> - Compute the score level by dividing each parameter score by 2 (using integer division). - Classify as: <b>Low</b> if score level is in {1,2}, <b>Medium</b> if in {3,4}, and <b>High</b> otherwise.	Score Level = $\left\lfloor \frac{S}{2} \right\rfloor$ , $\begin{cases} \text{Low} & \text{if } S \in \{1, 2\}, \\ \text{Medium} & \text{if } S \in \{3, 4\}, \\ \text{High} & \text{if } S \geq 5. \end{cases}$
9	<b>Update Association Matrix:</b> - For each parameter ( $p_1$ to $p_6$ ) and for each unique issue type in the responses, extract the corresponding values. - Divide these values by 2, compute their mean, round the result to obtain an integer, and update the association matrix.	For a parameter: $v_i = \frac{\text{value}}{2}$ , $\bar{v} = \frac{1}{n} \sum_{i=1}^n v_i$ , Final Value = round( $\bar{v}$ )
10	<b>Submit Responses and Save Data:</b> - On clicking the Submit Responses button, display a success message. - Append the new response entry to the DataFrame and update <code>response.csv</code> . - Display the last 5 responses and count today's respondents.	-
11	<b>Display Updated Association Matrix:</b> - Call the function to update <code>am.csv</code> based on the latest responses, and display it as a table.	-

1	parameter	anxiety	bipolar	depression	normal	ptsd
2	self acceptance	3	2	1	6	2
3	positive relations with others	3	4	2	6	3
4	autonomy	2	3	2	4	1
5	environmental mastery	2	4	1	6	4
6	purpose in life	4	4	1	6	3
7	personal growth	3	3	1	6	4

Sample Association Matrix

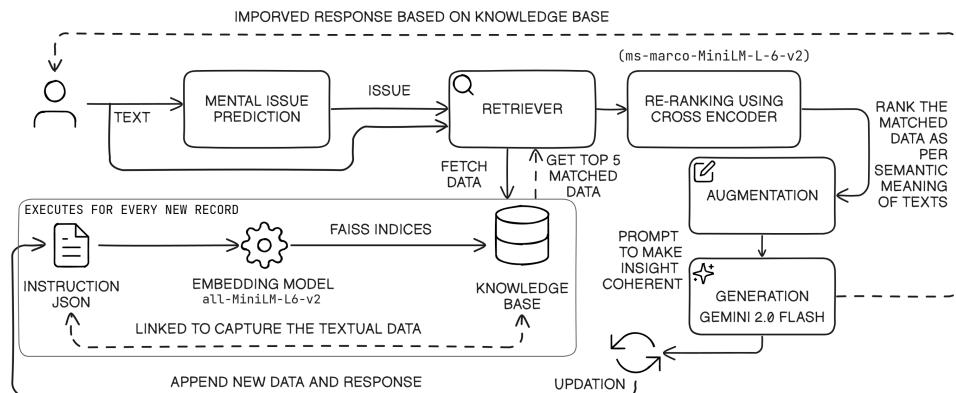
## Step-by-Step Algorithm for Association Matrix Analysis

Step	Operation	Mathematical Formula / Description
1	<b>Load Dataset</b>	Read the CSV file (e.g., am.csv) to obtain the association matrix data.
2	<b>Define Row Names and Issue Columns</b>	Set row names: self acceptance, positive relations with others, autonomy, environmental mastery, purpose in life, personal growth
3	<b>Extract Association Matrix</b>	Let $A \in \mathbb{R}^{6 \times 5}$ , $A_{ij}$ is the value for the $i$ -th parameter and $j$ -th issue with issue columns ordered as: anxiety, bipolar, depression, normal, ptsd.
4	<b>Define Probabilities Vector</b>	Define the probability vector where $I_j$ refers to the mental issue probabilities. $\vec{p} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix}, \text{ which satisfies } \sum_{j=1}^5 I_j = 1.$
5	<b>Weighted Sum Analysis</b>	Compute the weighted sum for each row: $w_i = \sum_{j=1}^5 A_{ij} p_j, \quad i = 1, \dots, 6.$ Identify the row index with maximum $w_i$ , i.e., $i^* = \arg \max_i w_i.$
6	<b>Cosine Similarity Analysis</b>	For each row vector $\vec{a}_i$ of $A$ , compute: $s_i = \frac{\vec{a}_i \cdot \vec{p}}{\ \vec{a}_i\  \ \vec{p}\ }, \quad i = 1, \dots, 6.$ Determine $i^* = \arg \max_i s_i.$
7	<b>Euclidean Distance Analysis</b>	For each row, compute the Euclidean distance: $d_i = \sqrt{\sum_{j=1}^5 (A_{ij} - p_j)^2}, \quad i = 1, \dots, 6.$ Find the row index with the smallest distance: $i^* = \arg \min_i d_i.$
8	<b>Consensus Decision</b>	Compare the row names identified by the three analyses. If all three methods return the same row name, that is the consensus; otherwise, list the unique names obtained.

### Step-by-Step Algorithm for Association Matrix Analysis

<b>Step</b>	<b>Operation</b>	<b>Mathematical Formula / Description</b>
9	<b>Display Results</b>	<p>Use Streamlit to display:</p> <ul style="list-style-type: none"> <li>• The original Association Matrix.</li> <li>• The computed weighted sums, cosine similarity scores, and Euclidean distances.</li> <li>• The row(s) corresponding to the maximum weighted sum, highest cosine similarity, and smallest Euclidean distance.</li> <li>• The consensus result.</li> </ul>

## 8.7 RAG for Wellbeing Insights



RAG OVERVIEW FOR GENERATING INSIGHTS

### Step-by-Step Algorithm for RAG-based Wellbeing Insight Generation

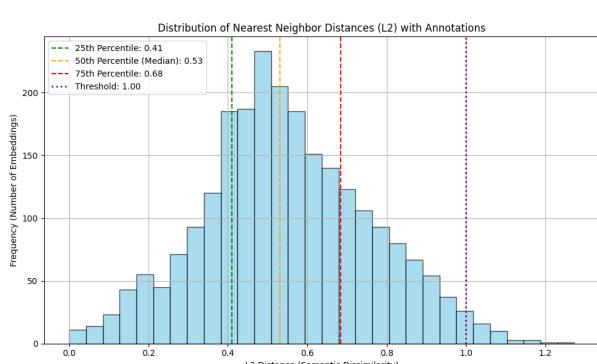
<b>Step</b>	<b>Operation</b>	<b>Mathematical Formula / Description</b>
1	<b>Import Libraries</b>	Import necessary packages: streamlit, json, pickle, numpy, faiss, sentence_transformers, transformers, google.generativeai, plotly, sklearn.manifold.TSNE, etc.
2	<b>Configure Gemini API</b>	Set the generation configuration parameters and initialize the Gemini model.
3	<b>Generate Instruction Dataset</b>	Create a dataset <code>instruction_data.json</code> with <code>text</code> , <code>issue</code> , and <code>response</code> using Gemini API through batching and rate limiting.

## Step-by-Step Algorithm for RAG-based Wellbeing Insight Generation

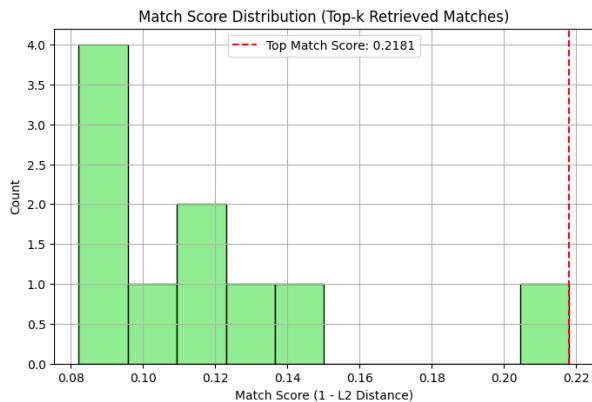
<b>Step</b>	<b>Operation</b>	<b>Mathematical Formula / Description</b>
4	<b>Build FAISS Index</b>	Load the saved model via: <code>embedding_model = pickle.load("rag_embedding_gpu.pkl")</code> . Then load the instruction data from <code>instruction_data.json</code> to form <code>documents = {instruction + input}</code> and <code>outputs = {output}</code> . Next, encode the documents using <code>embeddings = np.array([embedding_model.encode(doc)])</code> so that <code>embeddings ∈ ℝ<sup>n×d</sup></code> . Build the FAISS L2-index by setting <code>index = faiss.IndexFlatL2(d)</code> and calling <code>index.add(embeddings)</code> . Finally, save the tuple <code>{documents, outputs, index, embeddings}</code> into <code>"global_store_gpu.pkl"</code> .
5	<b>Load Global Store</b>	Retrieve data from <code>global_store_gpu.pkl</code> (documents, indices, outputs and embeddings).
6	<b>Compute Query Embedding</b>	Create the query by concatenating user text and mental issue. Encode it and cast to a float32 NumPy array: $v_q = \text{embedding\_model.encode(query)}$
7	<b>FAISS Retrieval (Top-<math>k</math>)</b>	Retrieve the top $k = 10$ matches using FAISS. The distance metric is the squared Euclidean (L2) norm: $d(v_q, v_i) = \ v_q - v_i\ ^2$ The query is processed as: $(D, I) = \text{index.search}(v_q, 5)$
8	<b>Re-ranking</b>	Re-rank the top $n = 5$ from the initially retrieved $k$ documents to improve relevance using <code>ms-marco-MiniLM-L-6-v2</code> . <ul style="list-style-type: none"> <li>(i) Extract text for each candidate: <code>retrieved_docs_text = {documents[i]   i ∈ initial_indices}</code>.</li> <li>(ii) Form pairs by concatenating the query with each document: <code>cross_inp = {[query, doc_text]}</code>.</li> <li>(iii) Compute cross-encoder scores: <code>cross_scores = cross_encoder.predict(cross_inp)</code>.</li> <li>(iv) Sort candidates by score in descending order</li> </ul>
9	<b>Filter by Ryff Parameters</b>	Apply text filtering to the retrieved output of the top candidate. Check if it contains any selected Ryff parameters (e.g., Autonomy, Positive Relations, etc.).
10	<b>Generate Refined Insight</b>	Construct a prompt for the Gemini model based on the retrieved text and selected parameters. Generate a refined insight: <code>generated_output = gemini_model.generate_content(prompt)</code>

## Step-by-Step Algorithm for RAG-based Wellbeing Insight Generation

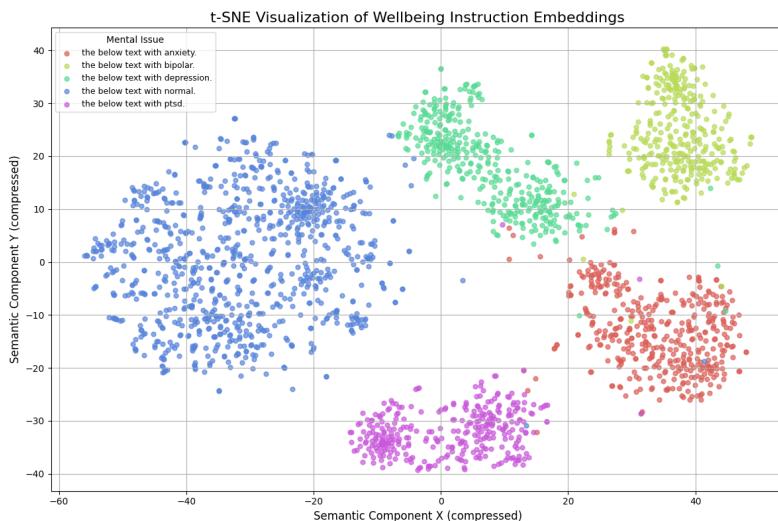
Step	Operation	Mathematical Formula / Description
11	<b>3D t-SNE Visualization</b>	Combine the query embedding with the document embeddings and reduce the dimensionality using t-SNE:  $\text{reduced} = \text{TSNE}(n\_components = 3)(\text{combined\_embeddings})$  Separate the query and top- $k$ points for visualization via Plotly.
12	<b>Plot Interactive Graph</b>	Plot the 3D scatter plot with annotations for the query, top- $k$ matches, and connecting dotted lines.
13	<b>Append New Record</b>	Append a new record to <code>instruction_data.json</code> and rebuild the FAISS index. In this step, the updated documents are:  $\text{documents} = \{\text{instruction} + \text{input}\}$  and the FAISS index is rebuilt with the new embeddings.



Nearest Neighbours among the Embeddings

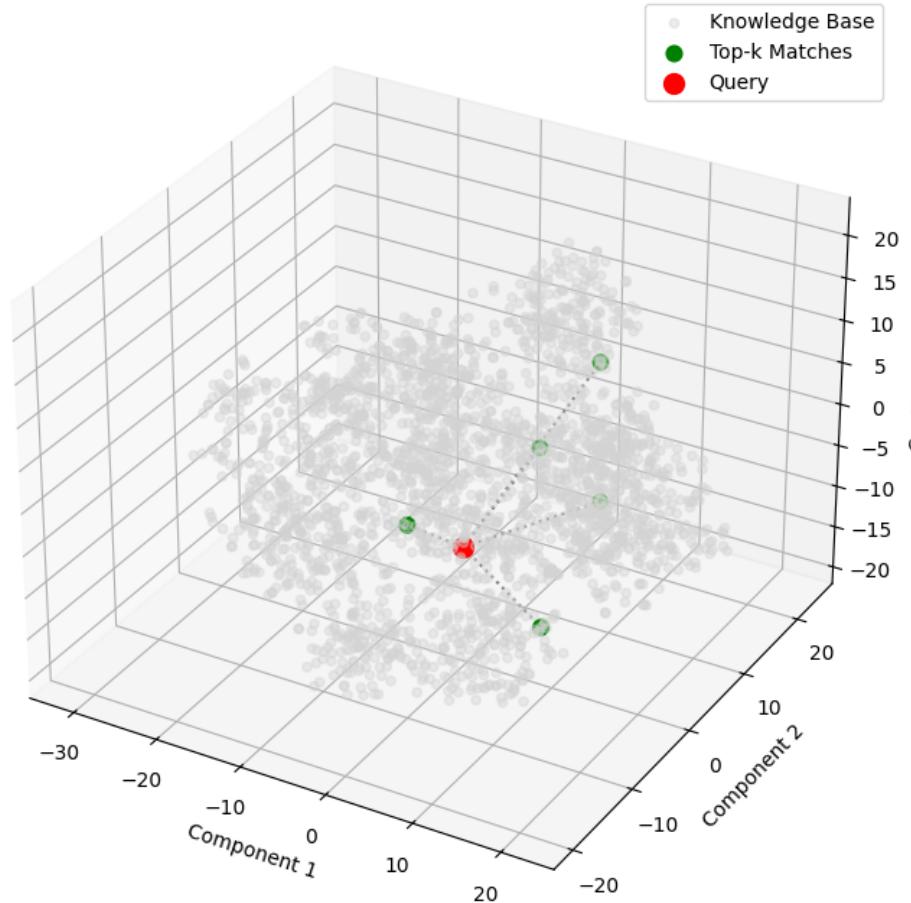


Count of Matches and Top Score for an input



Instruction Embeddings in 2D space

3D t-SNE: Query, Top-k Matches &amp; Knowledge Base

Visualization of Query and Top-*k* Matches

## 9 Test Plans, Results and Analysis

Test Case Plan Table

Test Case ID	Description	Expected Result	Actual Result	Status
T01	User provides input in the text area.	Text is accepted for further processing.	Text is accepted for further processing.	✓
T02	Display mental health issues with probabilities for classes.	Probabilities for each mental health class are shown.	Probabilities for each mental health class are shown.	✓
T03	Highlight the mental health issue with the highest probability.	Correct issue with the highest probability is displayed.	Correct issue with the highest probability is displayed.	✓
T04	Accept username input and provide prediction.	Prediction is displayed based on the provided username.	Prediction is displayed based on the provided username.	✓

## Test Case Plan Table

<b>Test Case ID</b>	<b>Description</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>
T05	Translate multiple language inputs to English.	Non-English text is translated correctly to English.	Non-English text is translated correctly to English.	✓
T06	Extract text and detect emotion from image.	Extracted text and detected emotions are displayed accurately.	Extracted text and detected emotions are displayed accurately.	✓
T07	Pass a prompt to the system and retrieve a valid response.	Correct response is generated based on the prompt.	Correct response is generated based on the prompt.	✓
T08	Perform a combined test using multiple inputs.	Predictions for all inputs are displayed correctly.	Predictions for all inputs are displayed correctly.	✓
T09	Analyze uploaded audio files and transcribe them into text.	Audio transcription and analysis results are displayed.	Audio transcription and analysis results are displayed.	✓
T10	Extract frames, analyze emotions, audio from video.	Frame emotions and audio transcription are displayed.	Frame emotions and audio transcription are displayed.	✓
T11	Extract tweets and related media using a Twitter username.	Text and media are extracted and analyzed correctly.	Text and media are extracted and analyzed correctly.	✓
T12	Generate captions for uploaded images or video frames.	Captions are generated for images or frames.	Captions are generated for images or frames.	✓
T13	Upload a PDF file and analyze its text content.	Extracted text from the PDF is processed and analyzed for mental health cues.	Extracted text from the PDF is processed and analyzed for mental health cues.	✓
T14	Capture user response to a displayed image.	User response is captured and correctly classified for emotional analysis.	User response is captured and correctly classified for emotional analysis.	✓
T15	Fill out a survey form to update the association matrix.	Survey responses update the association matrix and generate targeted wellbeing insights.	Survey responses update the association matrix and generate targeted wellbeing insights.	✓
T16	RAG-based Wellbeing Insights	Synthesizes targeted wellbeing insights by leveraging retrieved context, input prompts, and mental health cues.	Generates refined wellbeing insights using GEMINI-2.0-FLASH based on retrieved data.	✓

## Metrics for Evaluation

Metric	Definition and Formula
Precision	The ratio of correctly predicted positive observations to the total predicted positives. $\text{Precision} = \frac{TP}{TP + FP}$ where $TP$ = True Positives, $FP$ = False Positives.
Recall	The ratio of correctly predicted positive observations to all observations in the actual class. $\text{Recall} = \frac{TP}{TP + FN}$ where $TP$ = True Positives, $FN$ = False Negatives.
F1-Score	The harmonic mean of Precision and Recall. $\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
Support	The number of actual occurrences of each class in the dataset. $\text{Support} = \text{Number of samples in the true class}$
Confusion Matrix	A matrix used to evaluate classification performance: $\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$ where $TP$ = True Positives, $FP$ = False Positives, $FN$ = False Negatives, and $TN$ = True Negatives.

## 9.1 Results from Base Models

## Logistic Regression

Class	Precision	Recall	F1-Score	Support
Anxiety	0.83	0.77	0.80	379
Bipolar	0.74	0.55	0.63	384
Depression	0.76	0.76	0.76	373
Normal	0.92	0.99	0.95	2183
PTSD	0.87	0.77	0.82	394
<b>Accuracy</b>	87.66%			
<b>Macro Avg</b>	0.82	0.77	0.79	3713
<b>Weighted Avg</b>	0.87	0.88	0.87	3713

**Naive Bayes**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.70	0.73	0.72	379
Bipolar	0.83	0.45	0.58	384
Depression	0.59	0.87	0.70	373
Normal	0.96	0.92	0.94	2183
PTSD	0.71	0.83	0.76	394
<b>Accuracy</b>	83.63%			
<b>Macro Avg</b>	0.76	0.76	0.74	3713
<b>Weighted Avg</b>	0.85	0.84	0.84	3713

**Support Vector Machine**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.72	0.76	0.74	379
Bipolar	0.62	0.61	0.61	384
Depression	0.74	0.71	0.72	373
Normal	0.94	0.95	0.95	2183
PTSD	0.78	0.74	0.76	394
<b>Accuracy</b>	85.13%			
<b>Macro Avg</b>	0.76	0.75	0.76	3713
<b>Weighted Avg</b>	0.85	0.85	0.85	3713

**Random Forest**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.81	0.70	0.75	379
Bipolar	0.93	0.47	0.62	384
Depression	0.72	0.77	0.74	373
Normal	0.88	1.00	0.93	2183
PTSD	0.92	0.74	0.82	394
<b>Accuracy</b>	86.00%			
<b>Macro Avg</b>	0.85	0.73	0.77	3713
<b>Weighted Avg</b>	0.86	0.86	0.85	3713

**XGBoost**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.81	0.74	0.77	403
Bipolar	0.77	0.62	0.69	397
Depression	0.72	0.81	0.76	387
Normal	0.93	0.98	0.95	2137
PTSD	0.86	0.75	0.80	396
<b>Accuracy</b>	87.39%			
<b>Macro Avg</b>	0.82	0.78	0.80	3720
<b>Weighted Avg</b>	0.87	0.87	0.87	3720

**K-Nearest Neighbour**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.58	0.31	0.40	379
Bipolar	0.18	0.59	0.28	384
Depression	0.47	0.39	0.43	373
Normal	0.79	0.69	0.73	2183
PTSD	0.80	0.09	0.16	394
<b>Accuracy</b>	54.46%			
<b>Macro Avg</b>	0.56	0.41	0.40	3713
<b>Weighted Avg</b>	0.67	0.54	0.56	3713

**Long Short Term Memory**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.74	0.72	0.73	1999
Bipolar	0.66	0.70	0.68	1964
Depression	0.68	0.69	0.69	1959
Normal	0.97	0.94	0.95	10688
PTSD	0.72	0.78	0.75	1987
<b>Accuracy</b>	84.91%			
<b>Macro Avg</b>	0.75	0.77	0.76	18597
<b>Weighted Avg</b>	0.85	0.85	0.85	18597

**Custom Transformer**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.75	0.78	0.76	379
Bipolar	0.76	0.69	0.73	384
Depression	0.82	0.72	0.77	373
Normal	0.95	0.97	0.96	2183
PTSD	0.79	0.80	0.79	394
<b>Accuracy</b>	88.5%			
<b>Macro Avg</b>	0.81	0.79	0.80	3713
<b>Weighted Avg</b>	0.88	0.88	0.88	3713

**Hyperparameter Tuning on Logistic Regression**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.83	0.77	0.80	379
Bipolar	0.74	0.55	0.64	384
Depression	0.76	0.76	0.76	373
Normal	0.92	0.99	0.95	2183
PTSD	0.87	0.77	0.82	394
<b>Accuracy</b>	87.72%			
<b>Macro Avg</b>	0.83	0.77	0.79	3713
<b>Weighted Avg</b>	0.87	0.88	0.87	3713

**Hyperparameter Tuning on KNN**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.73	0.23	0.35	379
Bipolar	0.18	0.60	0.27	384
Depression	0.47	0.40	0.43	373
Normal	0.74	0.65	0.69	2183
PTSD	0.83	0.10	0.18	394
<b>Accuracy</b>	52.03%			
<b>Macro Avg</b>	0.59	0.40	0.39	3713
<b>Weighted Avg</b>	0.66	0.52	0.53	3713

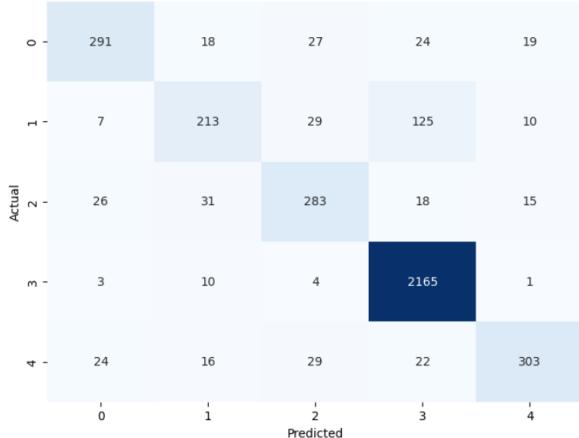
**Hyperparameter Tuning on SVM**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.72	0.76	0.74	379
Bipolar	0.62	0.61	0.61	384
Depression	0.74	0.71	0.72	373
Normal	0.94	0.95	0.95	2183
PTSD	0.78	0.74	0.76	394
<b>Accuracy</b>	85.13%			
<b>Macro Avg</b>	0.76	0.75	0.76	3713
<b>Weighted Avg</b>	0.85	0.85	0.85	3713

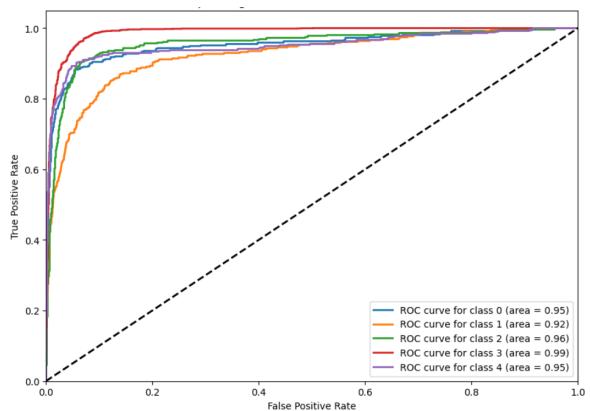
**Hyperparameter Tuning on Naive Bayes**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.69	0.76	0.72	379
Bipolar	0.75	0.55	0.64	384
Depression	0.60	0.83	0.70	373
Normal	0.96	0.91	0.94	2183
PTSD	0.73	0.79	0.76	394
<b>Accuracy</b>	83.95%			
<b>Macro Avg</b>	0.75	0.77	0.75	3713
<b>Weighted Avg</b>	0.85	0.84	0.84	3713

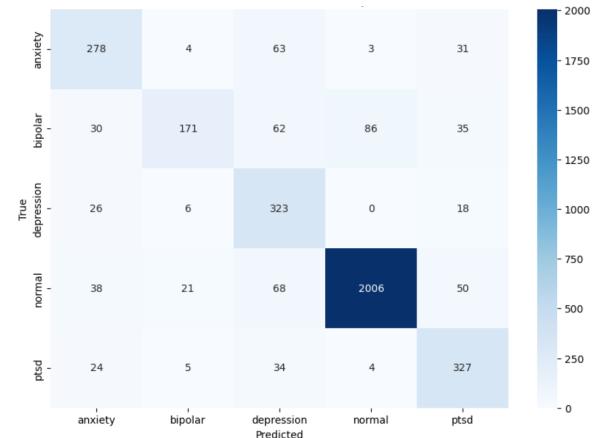
# MAFSMBMDDPW



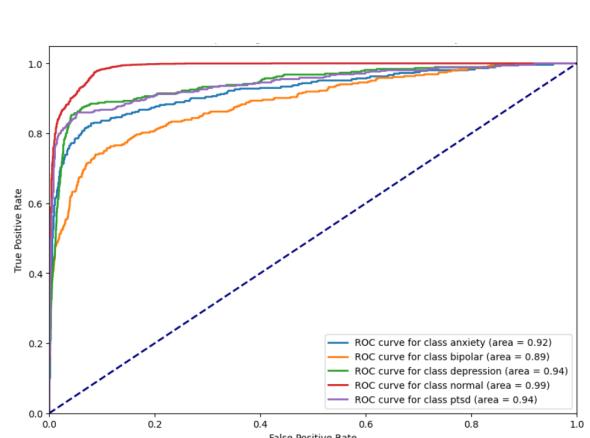
Confusion Matrix (Logistic Regression)



ROC AUC (Logistic Regression)



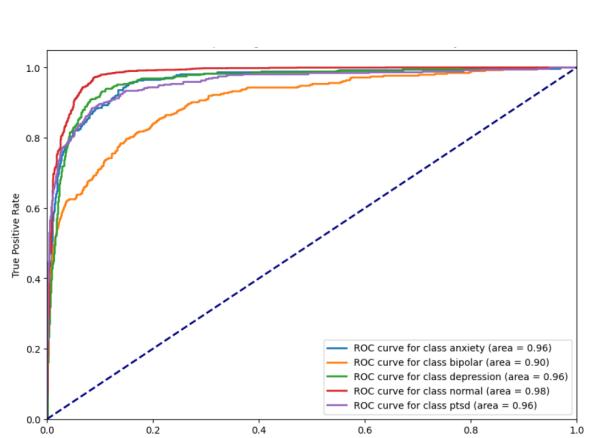
Confusion Matrix (Naive Bayes)



ROC AUC (Naive Bayes)

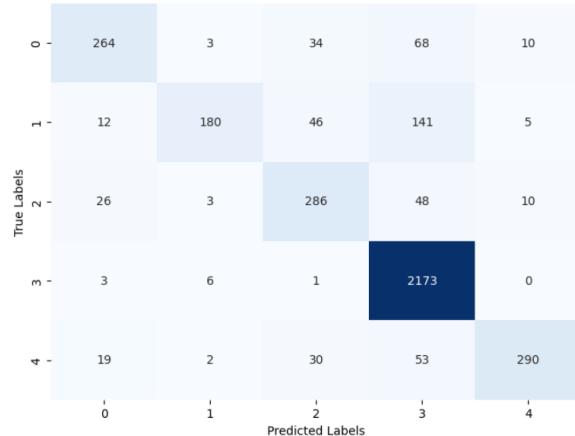


Confusion Matrix (SVM)

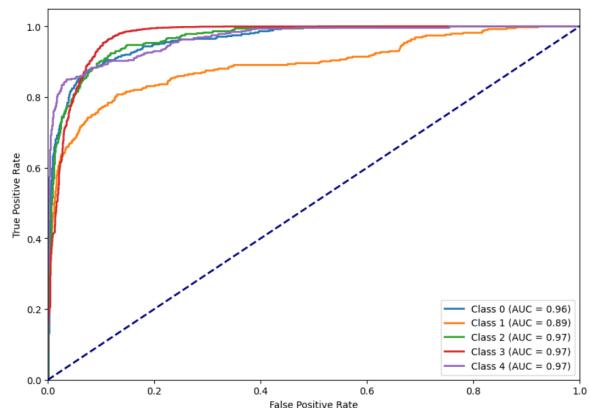


ROC AUC (SVM)

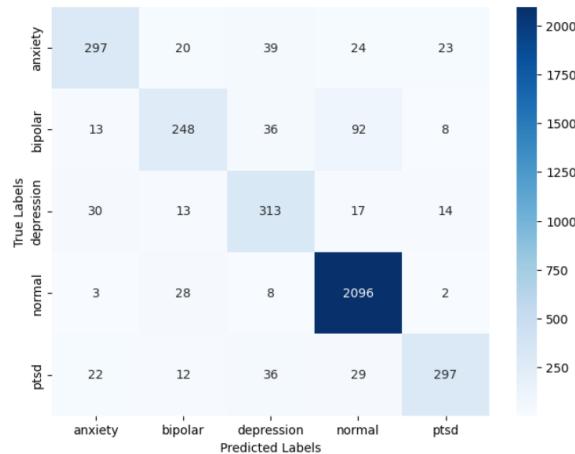
# MAFSMBMDDPW



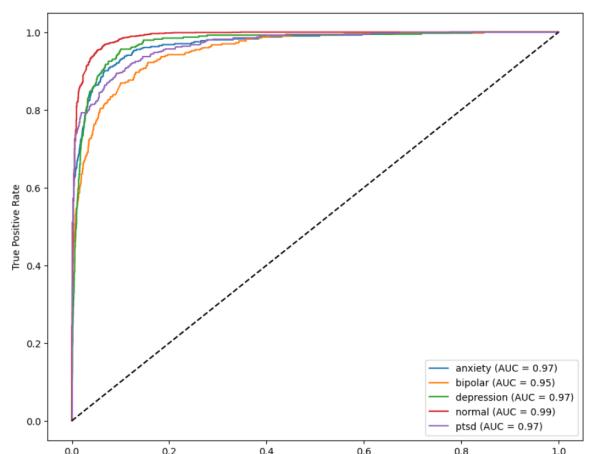
Confusion Matrix (Random Forest)



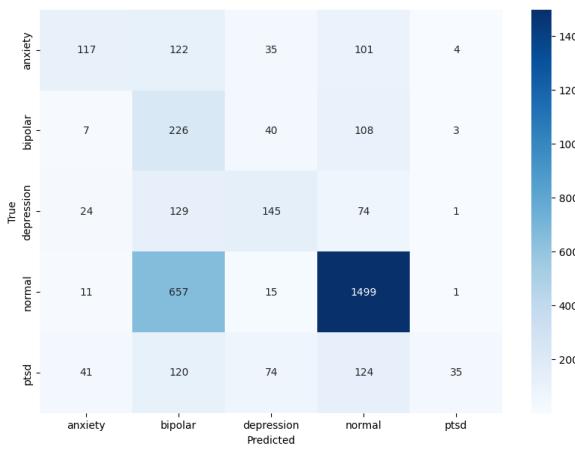
ROC AUC (Random Forest)



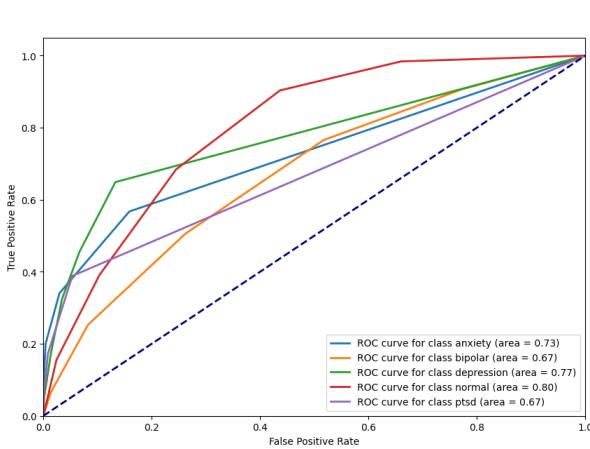
Confusion Matrix (XGBoost)



ROC AUC (XGBoost)

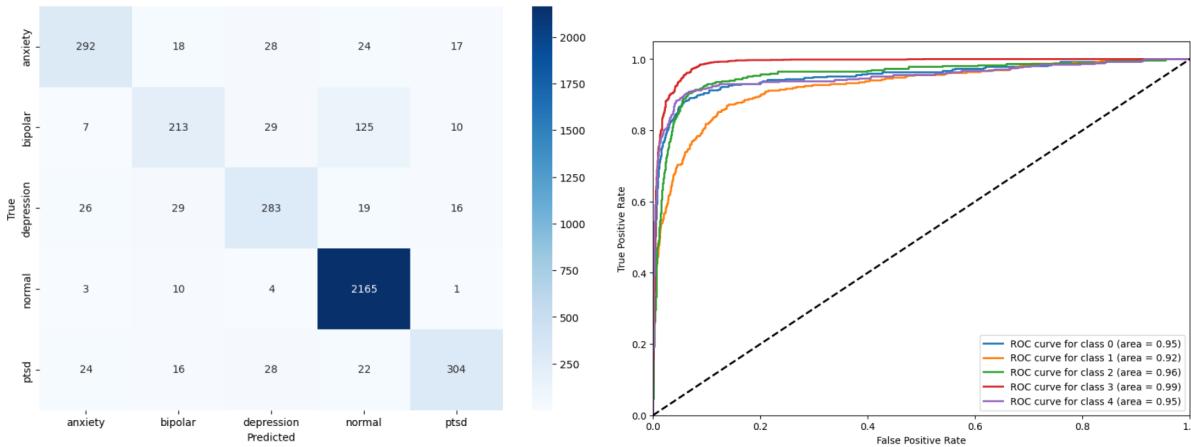
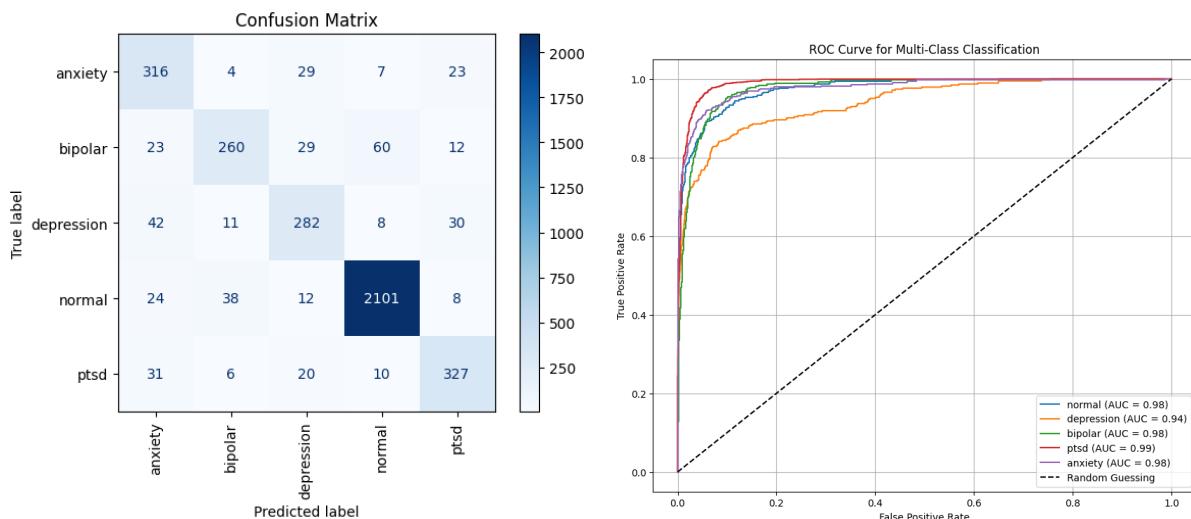
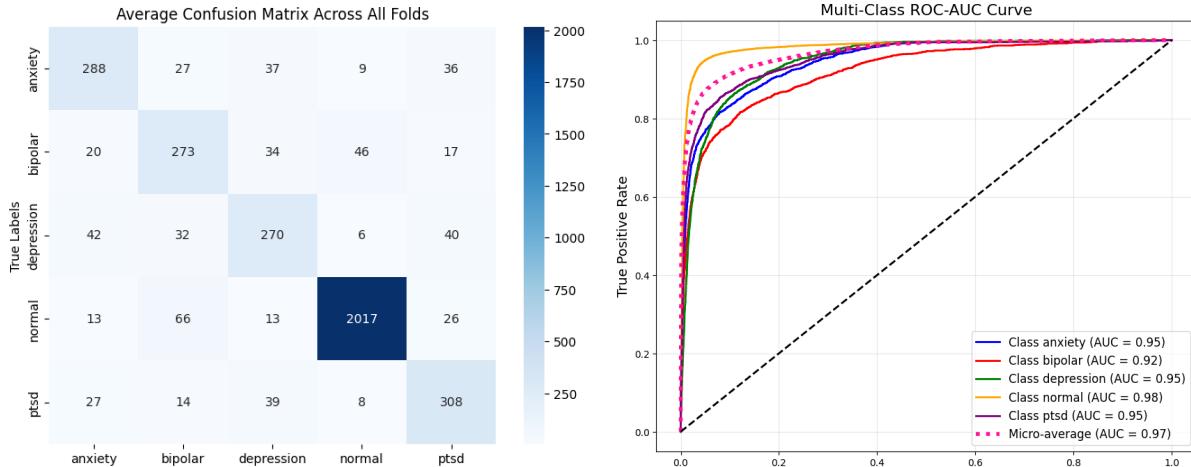


Confusion Matrix (KNN)

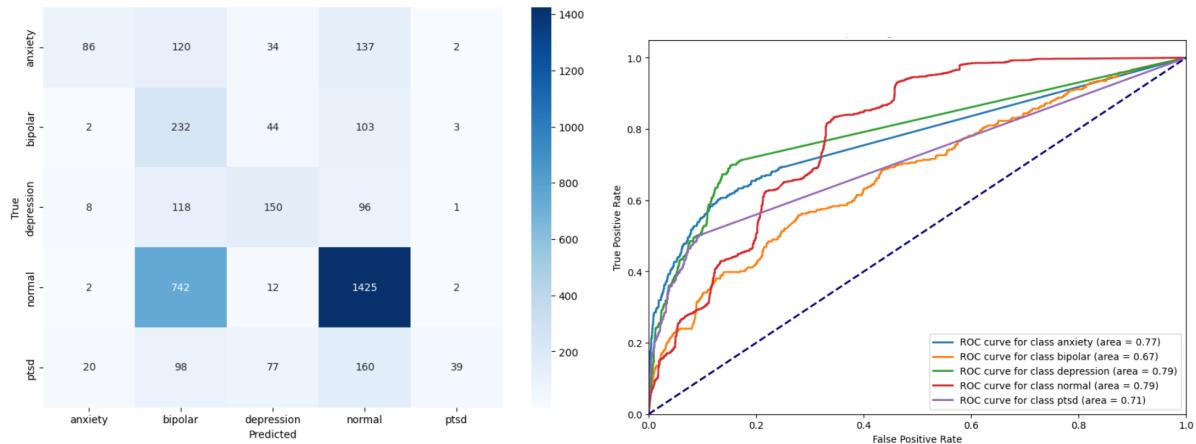


ROC AUC (KNN)

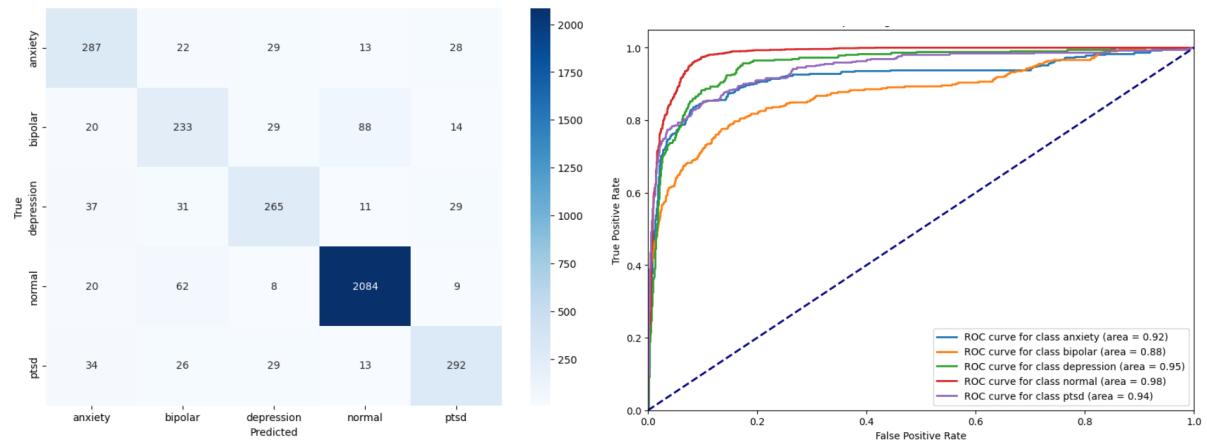
# MAFSMBMDDPW



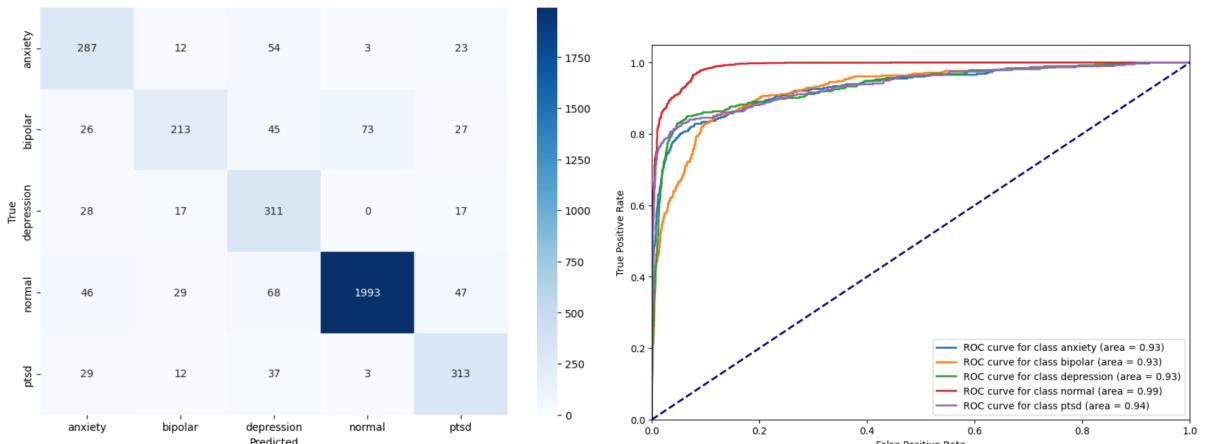
## MAFSMBMDDPW



Confusion Matrix and ROC AUC for KNN with Hyperparameter Tuning



Confusion Matrix and ROC AUC for SVM with Hyperparameter Tuning



Confusion Matrix and ROC AUC for Naive Bayes with Hyperparameter Tuning

Model	Accuracy (%)	Key Observations
Logistic Regression	87.66	High precision, recall, and F1 for the <i>Normal</i> class; struggles with <i>Bipolar</i> and <i>Anxiety</i> . ROC AUC: Normal (0.99), Anxiety (0.95), Depression (0.96).
Naive Bayes	83.63	Excellent for <i>Normal</i> (Precision 0.96, Recall 0.92, F1 0.94) but poor for <i>Bipolar</i> (Recall 0.45, F1 0.58). ROC AUC: Normal (0.99), Depression/PTSD (0.94), Bipolar (0.89).
SVM	85.13	Best for <i>Normal</i> (Precision 0.94, Recall 0.95, F1 0.95); <i>Bipolar</i> underperforms (Precision 0.62, Recall 0.61); <i>Anxiety</i> is balanced (Recall 0.76, Precision 0.72). ROC AUC: Normal (0.98), others 0.96, Bipolar (0.90).
Random Forest	86.00	<i>Normal</i> class achieves perfect recall (1.00) with high precision (0.88, F1 0.93); <i>Bipolar</i> is weak (Recall 0.47, F1 0.62). ROC AUC: For most classes it is greater than equal to 0.96; Bipolar (0.89).
XGBoost	87.39	Strong performance for <i>Normal</i> ; <i>Anxiety</i> shows good metrics (Precision 0.81, Recall 0.74) while <i>Bipolar</i> has lower recall (0.62). ROC AUC: 0.97 for Anxiety, Depression, PTSD; 0.99 for Normal.
KNN	54.46	Overall low performance; <i>Normal</i> is moderate (Precision 0.79, Recall 0.69) but <i>PTSD</i> has very low recall (0.09). ROC AUC: Normal (0.80), indicating weak discrimination.
LSTM	84.91	High performance for <i>Normal</i> ; <i>Anxiety</i> acceptable (Precision 0.74, Recall 0.72); <i>Bipolar</i> (Precision 0.66, Recall 0.70) and <i>PTSD</i> (Precision 0.72, Recall 0.78) are moderate. Effective ROC AUC for Normal.
Transformer	88.50	Best overall with balanced high precision and recall across all classes. Captures contextual cues effectively and handles class imbalance well. ROC AUC scores range between 0.94 and 0.99.

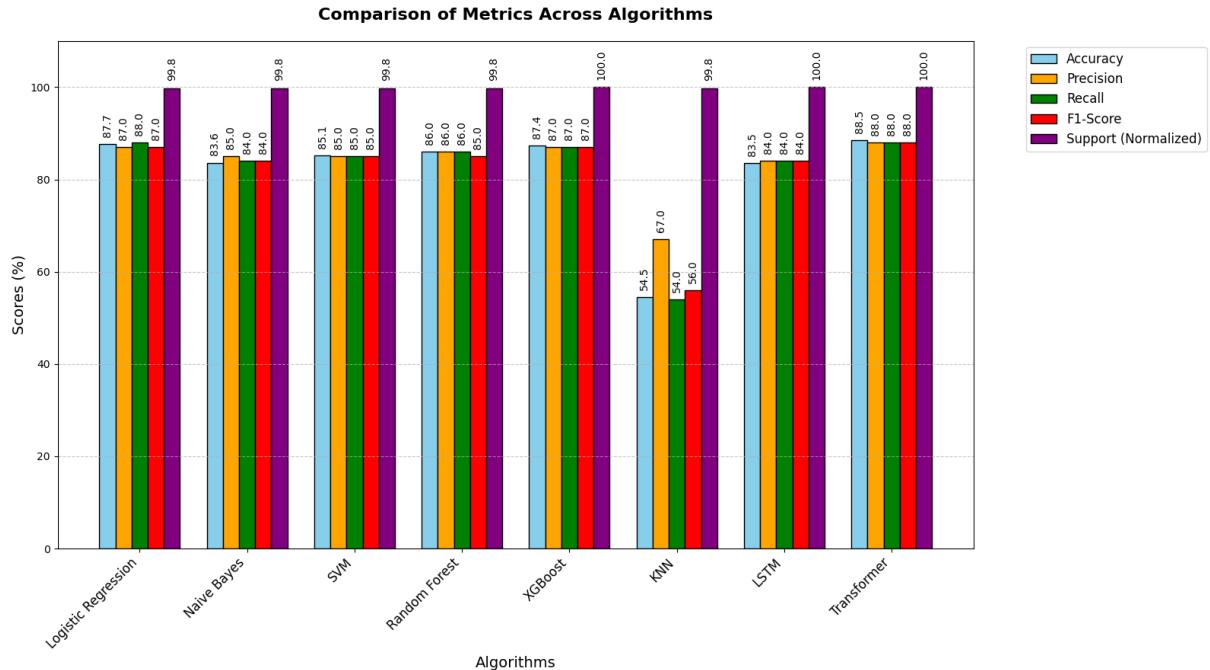
### Summary Comparison of Classification Models without Hyperparameter Tuning

Model	Best Hyperparameters	Accuracy & Metrics	Key Observations
Logistic Regression	solver=liblinear, penalty=l2, C=1	Accuracy: 87.72%; Normal: 92% accuracy; ROC AUC: Normal 0.99, Depression 0.96	High precision, recall, and F1 for Normal; Bipolar shows lower F1 (64%); overall robust performance, especially for Normal and Anxiety.
k-NN	weights=distance, n_neighbors=10, metric=euclidean	Accuracy: 52.03%; Normal F1: 0.69; ROC AUC: Bipolar 0.67, PTSD 0.71, Normal/Depression 0.79	Struggles with minority classes: Anxiety (recall 0.23) and PTSD (recall 0.10); not suitable for this dataset.
SVM	kernel=linear, gamma=scale, C=1	Accuracy: 85.13%; F1-scores: Normal 0.95, Depression 0.72, Anxiety 0.74, Bipolar 0.61; ROC AUC: Normal 0.98, Others >0.90, Bipolar 0.90	Solid overall performance; excellent for Normal and Anxiety, but slightly lower performance for Bipolar.
Naive Bayes	alpha=0.2914	Accuracy: 83.95%; F1-scores: Normal 0.94, PTSD 0.76, Anxiety 0.72; ROC AUC: Normal 0.99, Others >0.90	Performs very well for Normal and PTSD; lower scores for Bipolar and Depression.

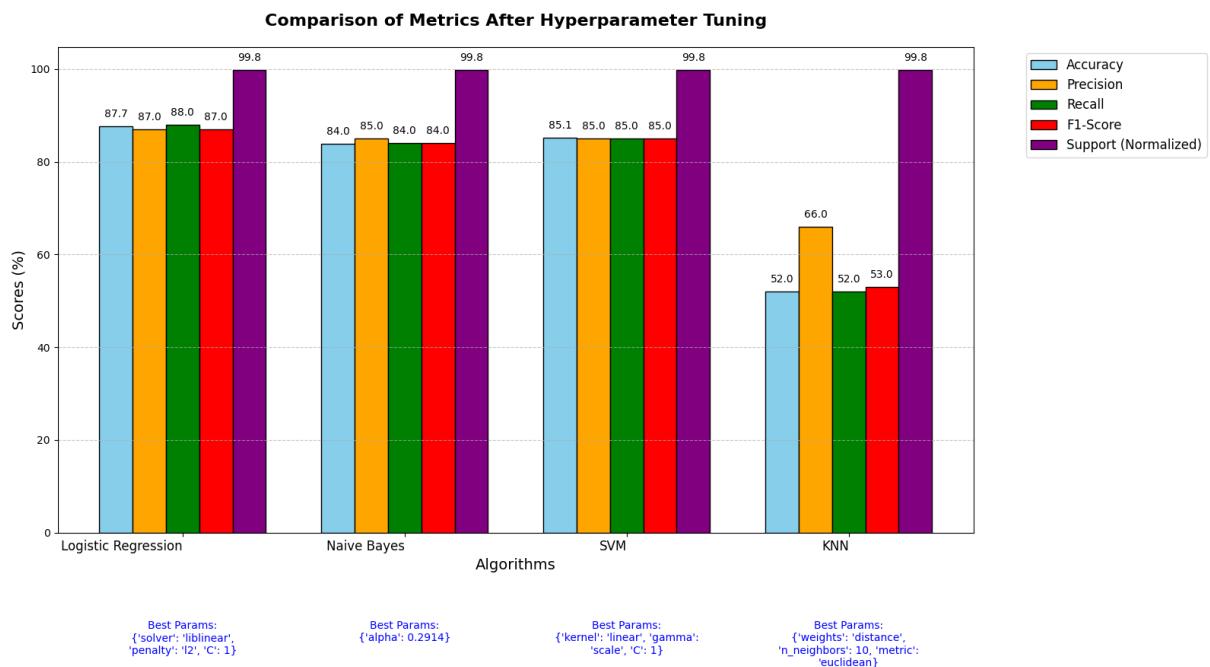
### Summary Comparison of Models after Hyperparameter Tuning

K-Nearest Neighbors (KNN) performs poorly on this mental health dataset compared to other algorithms, even after hyperparameter tuning. KNN relies on distance-based metrics, which struggle with the sparse, high-dimensional nature of text data, making it difficult to capture meaningful relationships between Reddit posts and mental health labels. Despite optimizing parameters like `weights='distance'`, `n_neighbors=10`, and `metric='euclidean'`, KNN's performance remains suboptimal, with lower accuracy, precision, recall, and F1-score than Logistic Regression, Naive Bayes, or SVM. Its inability to handle non-linear, context-dependent patterns in text, especially for nuanced categories like anxiety, PTSD, or bipolar disorder, highlights its limitations in text classification tasks. KNN's reliance on proximity without considering textual context likely explains its poor performance.

# MAFSMBMDDPW



**Result Comparison of the Algorithms**



**Result Comparison after Hyperparameter Tuning**

## 9.2 Comparison of different tokenizations

**Model Performance Comparison**

<b>Model</b>	<b>BoW</b>	<b>TFIDF</b>	<b>LIWC</b>	<b>Word2Vec</b>	<b>N-Gram</b>
<b>Logistic Regression</b>	87.66	86.02	66.36	79.40	87.13
<b>KNN</b>	54.56	75.06	70.70	75.52	43.06
<b>SVC</b>	85.13	88.26	68.14	77.89	84.33
<b>Naive Bayes</b>	83.63	79.42	59.63	60.44	81.71
<b>Random Forest</b>	85.32	85.73	76.73	79.88	79.02
<b>XGB</b>	87.42	87.39	78.56	81.01	87.96

In ensemble learning, combining models like Logistic Regression (LR), Naive Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), XGBoost (XGB), Random Forest (RF), and Long Short-Term Memory (LSTM) poses challenges in computational efficiency, model size, and accuracy. Each model has unique strengths and limitations, making their integration complex. LR, NB, and SVM perform well with simpler vectorization methods like Bag of Words (BoW), which efficiently represents text data as sparse vectors. BoW's simplicity ensures these models remain computationally lightweight while maintaining good classification accuracy. In contrast, XGBoost benefits from more complex feature extraction methods like TFIDF (Term Frequency-Inverse Document Frequency), which captures nuanced relationships by weighting words based on their importance across documents. However, combining TFIDF-based models like XGBoost with BoW-based models can degrade ensemble performance, as the feature representations may not align well. KNN is excluded due to its computational intensity, as it requires storing the entire training dataset and performs poorly with large, high-dimensional datasets. Similarly, Random Forest is omitted because its multiple decision trees increase model size and inference time, outweighing its individual performance benefits. XGBoost, while highly accurate with N-Gram features, is computationally expensive and impractical for real-time systems or large-scale deployment. The choice of feature extraction methods—BoW for LR, NB, and SVM, and TFIDF for XGBoost—balances simplicity and complexity. BoW ensures efficient, interpretable representations for simpler models, while TFIDF provides richer, context-aware features for XGBoost. This combination leverages the strengths of both methods, ensuring each model operates effectively without excessive computational strain. Ultimately, this approach strikes a balance between accuracy and efficiency, enabling the ensemble to perform well while remaining scalable and practical for real-world applications.

### 9.3 Results from Ensemble Model Training and Testing

**Ensemble Model 1**

	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>Anxiety</b>	0.97	0.94	0.95	400
<b>Bipolar</b>	0.92	0.85	0.88	388
<b>Depression</b>	0.95	0.94	0.94	392
<b>Normal</b>	0.97	0.99	0.98	2136
<b>PTSD</b>	0.97	0.96	0.96	397
<b>Accuracy</b>	<b>96.08%</b>			
<b>Macro avg</b>	0.96	0.93	0.95	3713
<b>Weighted avg</b>	0.96	0.96	0.96	3713

**Ensemble Model 2**

	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>Anxiety</b>	0.97	0.97	0.97	400
<b>Bipolar</b>	0.95	0.92	0.93	388
<b>Depression</b>	0.98	0.96	0.97	392
<b>Normal</b>	0.99	0.99	0.99	2136
<b>PTSD</b>	0.97	0.98	0.98	397
<b>Accuracy</b>	<b>97.93%</b>			
<b>Macro avg</b>	0.97	0.97	0.97	3713
<b>Weighted avg</b>	0.98	0.98	0.98	3713

**Ensemble Model 3**

	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>Anxiety</b>	0.98	0.97	0.98	400
<b>Bipolar</b>	0.96	0.90	0.93	388
<b>Depression</b>	0.97	0.96	0.96	392
<b>Normal</b>	0.98	1.00	0.99	2136
<b>PTSD</b>	0.97	0.97	0.97	397
<b>Accuracy</b>	<b>97.76%</b>			
<b>Macro avg</b>	0.97	0.96	0.97	3713
<b>Weighted avg</b>	0.98	0.98	0.98	3713

**Ensemble Model 4**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
Anxiety	0.97	0.97	0.97	400
Bipolar	0.95	0.90	0.92	388
Depression	0.97	0.95	0.96	392
Normal	0.99	0.99	0.99	2136
PTSD	0.97	0.98	0.97	397
<b>Accuracy</b>	<b>97.63%</b>			
<b>Macro Avg</b>	0.97	0.96	0.96	3713
<b>Weighted Avg</b>	0.98	0.98	0.98	3713

**Ensemble Model 5**

	Precision	Recall	F1-Score	Support
<b>Anxiety</b>	0.96	0.95	0.96	400
<b>Bipolar</b>	0.94	0.89	0.92	388
<b>Depression</b>	0.96	0.95	0.95	392
<b>Normal</b>	0.98	0.99	0.99	2136
<b>PTSD</b>	0.97	0.97	0.97	397
<b>Accuracy</b>	<b>97.17%</b>			
<b>Macro avg</b>	0.96	0.95	0.96	3713
<b>Weighted avg</b>	0.97	0.97	0.97	3713

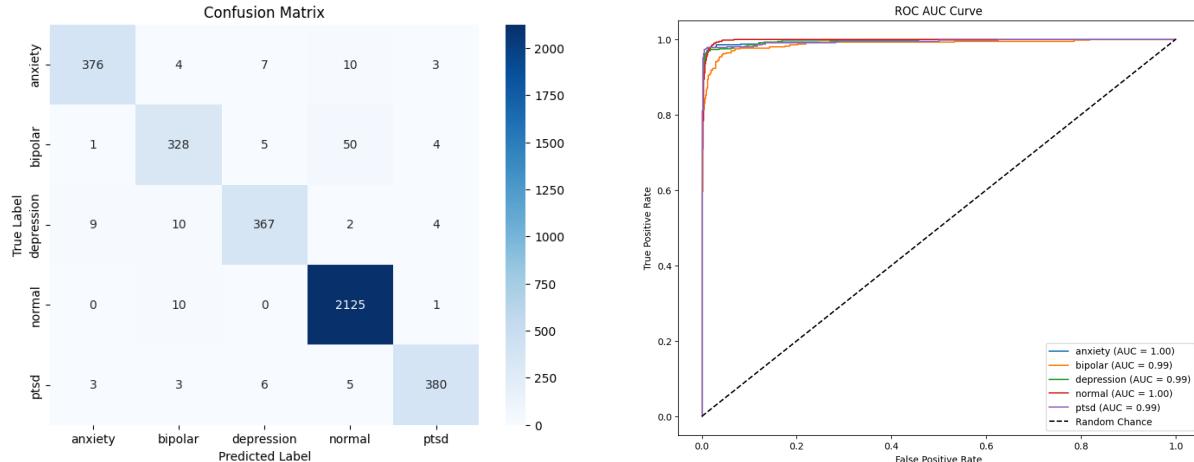
**Ensemble Model 6**

	Precision	Recall	F1-Score	Support
<b>Anxiety</b>	0.95	0.92	0.94	400
<b>Bipolar</b>	0.96	0.74	0.84	388
<b>Depression</b>	0.92	0.94	0.93	392
<b>Normal</b>	0.95	1.00	0.97	2136
<b>PTSD</b>	0.98	0.93	0.96	397
<b>Accuracy</b>	<b>95.15%</b>			
<b>Macro avg</b>	0.95	0.91	0.93	3713
<b>Weighted avg</b>	0.95	0.95	0.95	3713

**Ensemble Model 7**

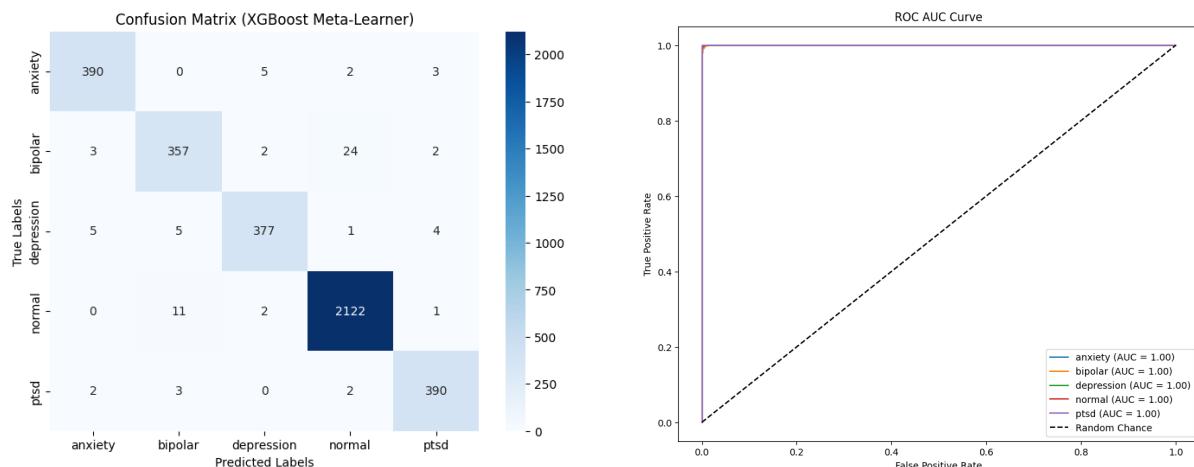
	Precision	Recall	F1-Score	Support
<b>Anxiety</b>	0.98	0.97	0.98	400
<b>Bipolar</b>	0.96	0.93	0.95	388
<b>Depression</b>	0.97	0.97	0.97	392
<b>Normal</b>	0.99	1.00	0.99	2136
<b>PTSD</b>	0.98	0.98	0.98	397
<b>Accuracy</b>	<b>98.03%</b>			
<b>Macro avg</b>	0.98	0.97	0.97	3713
<b>Weighted avg</b>	0.98	0.98	0.98	3713

# MAFSMBMDDPW



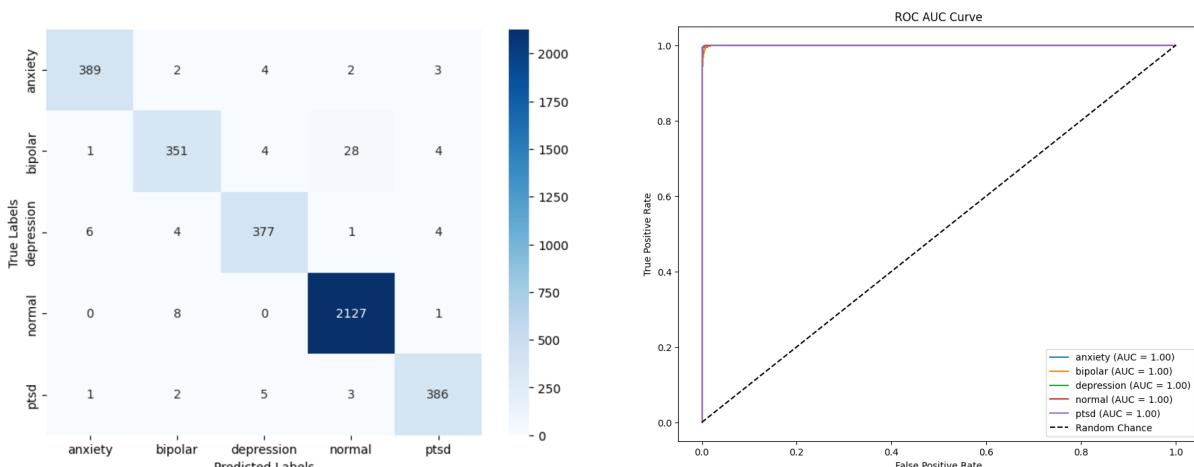
Confusion Matrix (Ensemble Model 1)

ROC AUC (Ensemble Model 1)



Confusion Matrix (Ensemble Model 2)

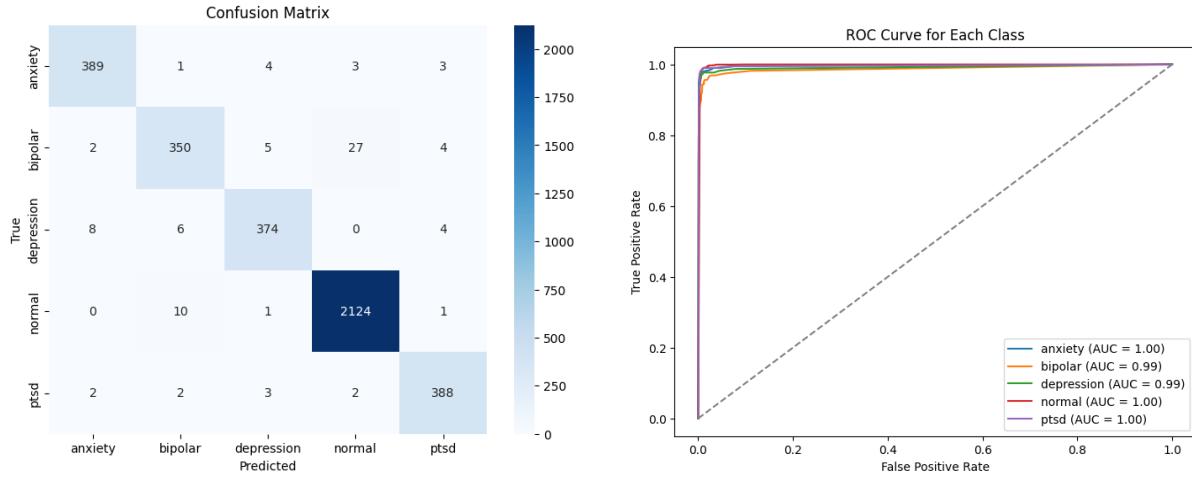
ROC AUC (Ensemble Model 2)



Confusion Matrix (Ensemble Model 3)

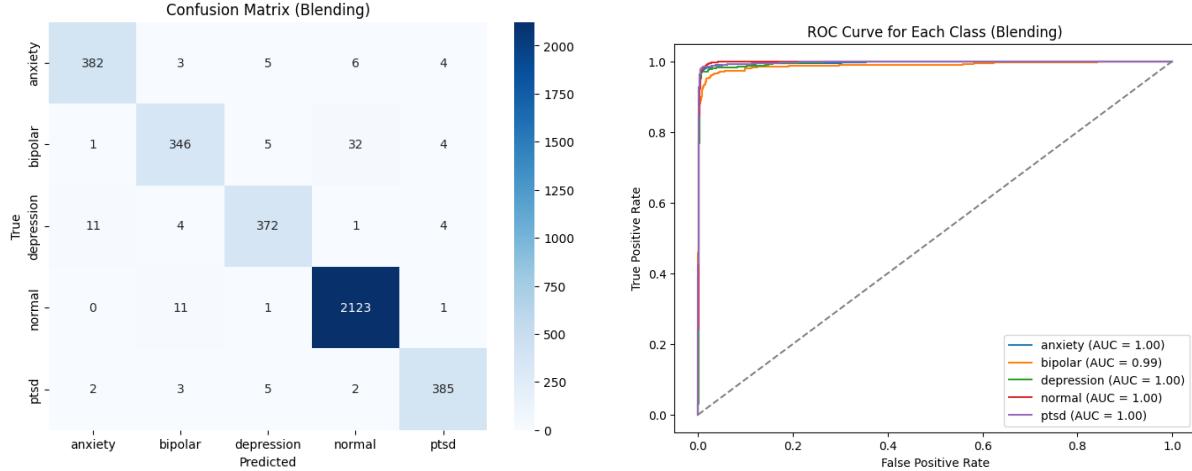
ROC AUC (Ensemble Model 3)

# MAFSMBMDDPW



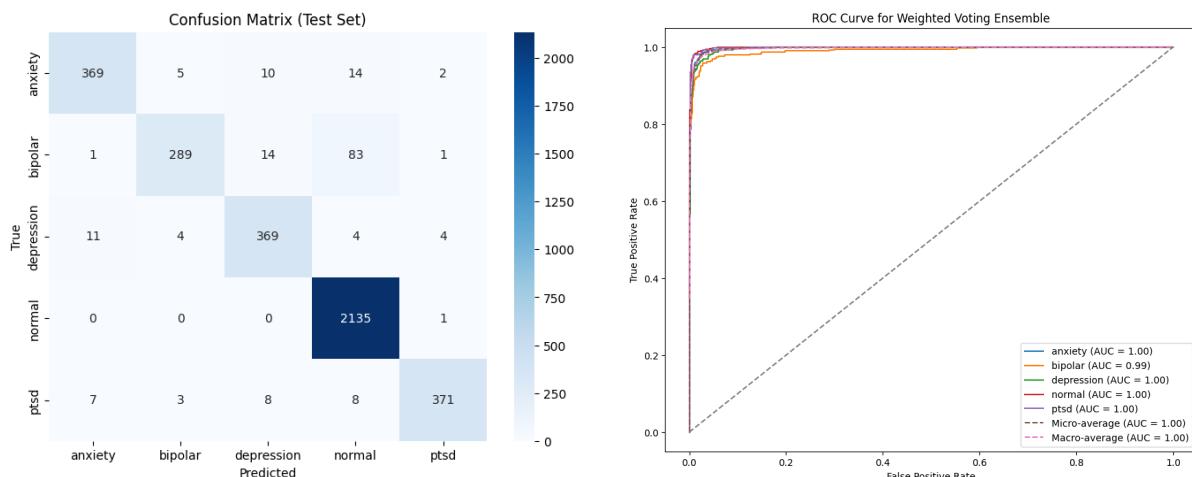
Confusion Matrix (Ensemble Model 4)

ROC AUC (Ensemble Model 4)



Confusion Matrix (Ensemble Model 5)

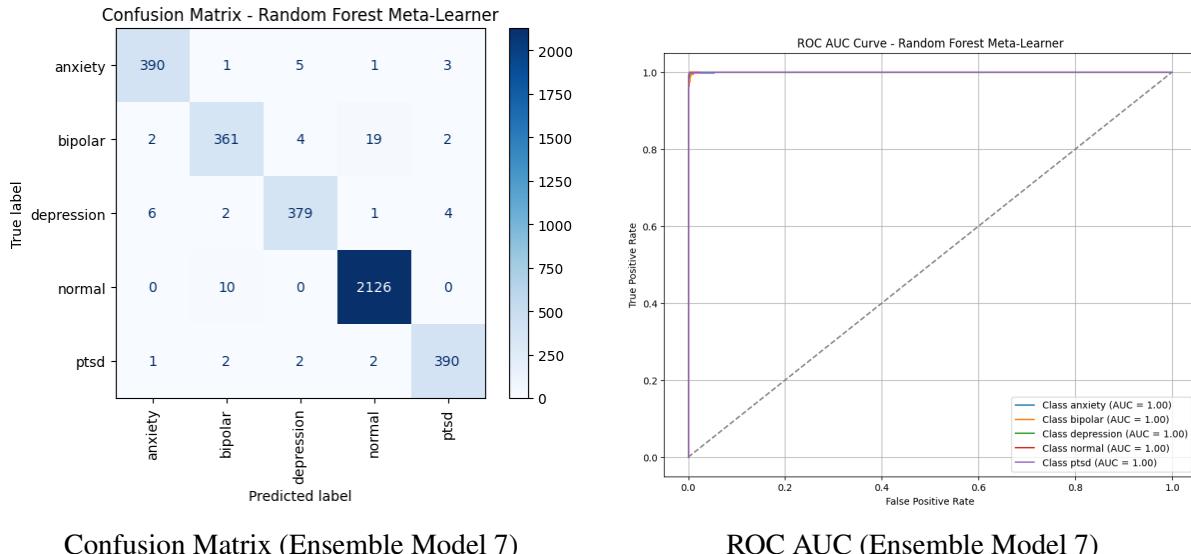
ROC AUC (Ensemble Model 5)



Confusion Matrix (Ensemble Model 6)

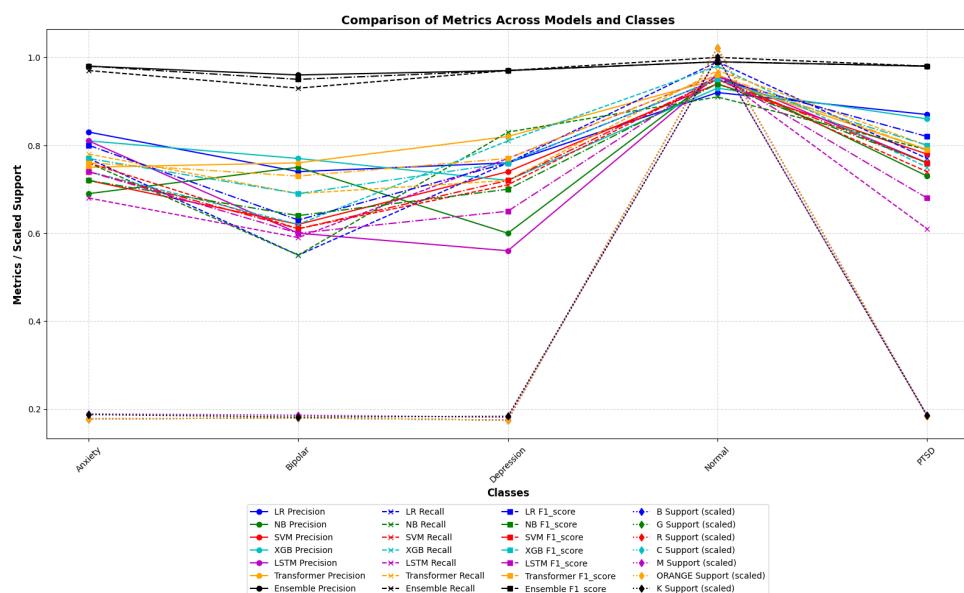
ROC AUC (Ensemble Model 6)

# MAFSMBMDDPW



Confusion Matrix (Ensemble Model 7)

ROC AUC (Ensemble Model 7)



Comparison of Base Models and Ensemble Model7

Comparison of all the 7 Ensemble Models

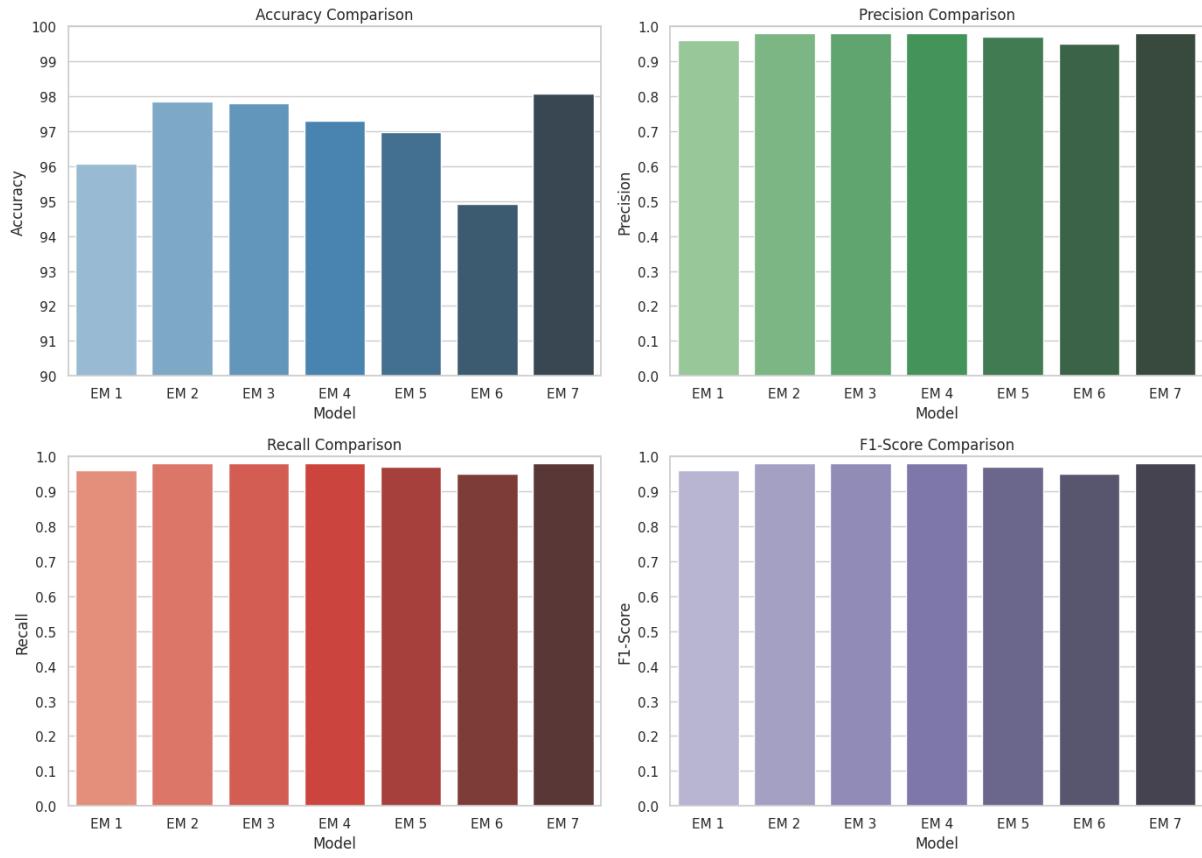
Model	Accuracy	Key Components	Advantages	Limitations
Ensemble Model 1	96.08%	Logistic Regression + XGBoost	Balanced performance (e.g., Anxiety: Prec=0.97, Rec=0.94, F1=0.95); robust macro/weighted averages	May not fully capture complex, non-linear, context-dependent patterns

## Comparison of all the 7 Ensemble Models

Model	Accuracy	Key Components	Advantages	Limitations
Ensemble Model 2	97.93	XGBoost as meta-learner combining base models (LR, SVM, NB, LSTM)	Near-perfect ROC AUC (1.0 for Anxiety, Normal, PTSD; 0.99 for Bipolar, Depression); effective handling of imbalanced data	Potential overfitting if base model predictions are highly correlated
Ensemble Model 3	97.76	Random Forest used as meta-learner	High overall accuracy with very few misclassifications; robust due to bootstrapping and random feature selection	Slightly lower recall in some classes (e.g., Bipolar)
Ensemble Model 4	97.63%	Bagging classifier (similar to Random Forest)	Excellent performance with near-perfect results for the “Normal” class	Tendency to overfit due to using all features without feature-level randomness
Ensemble Model 5	97.17%	Blending Meta-Learner trained directly on base model predictions	Strong precision, recall, and F1-scores across classes; stable cross-validation performance	More prone to overfitting compared to stacking ensembles (e.g., with a Random Forest meta-learner)
Ensemble Model 6	95.15%	Weighted Voting ensemble combining base model outputs	Solid overall performance with high recall for “Normal” and computational efficiency	Lacks optimal inter-model learning; struggles with distinguishing bipolar disorder
Ensemble Model 7	98.03%	Transformer-based model (base) + meta-learner (Random Forest and others)	Near-perfect ROC AUC (1.0 across classes); excellent accuracy and robust classification	Increased computational complexity due to Transformer integration

## MAFSMBMDDPW

Below is a comparison of all the ensemble models for reference



Comparison of all Ensemble Models

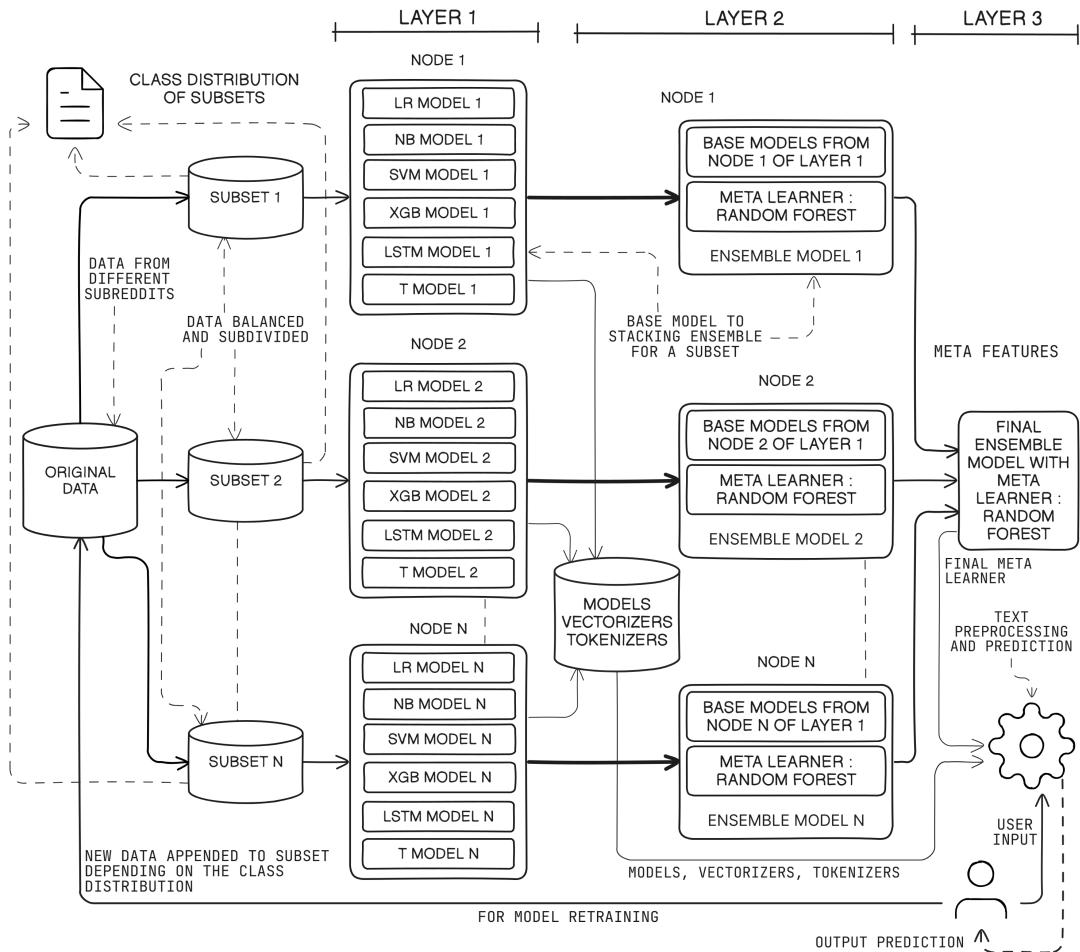
### 9.4 Results from hierarchical Ensemble Models

Ensemble Model 7 is applied on different subsets of a very large dataset to create hierarchical ensemble models. The performance of each model is evaluated, and the results are compared to the global ensemble model.

#### Performance Comparison of Models

Model	Subset 1	Subset 2	Subset 3	Subset 4	Subset 5	Subset 6	ENSEMBLE
Logistic Regression	87.66	85.07	86.74	90.61	88.17	72.62	<b>90.90</b>
Naive Bayes	83.63	81.71	83.50	84.44	85.93	64.33	<b>84.15</b>
SVM	85.13	82.30	83.50	88.34	85.45	67.81	<b>91.79</b>
XGBoost	87.39	85.79	86.78	91.37	86.32	70.95	<b>67.67</b>
LSTM	84.91	81.63	81.83	87.22	85.38	67.74	<b>87.48</b>
Transformer	88.50	84.50	86.50	89.35	87.62	72.40	<b>91.53</b>
<b>ENSEMBLE</b>	<b>98.03</b>	<b>94.85</b>	<b>96.96</b>	<b>97.79</b>	<b>96.86</b>	<b>92.57</b>	<b>96.24</b>

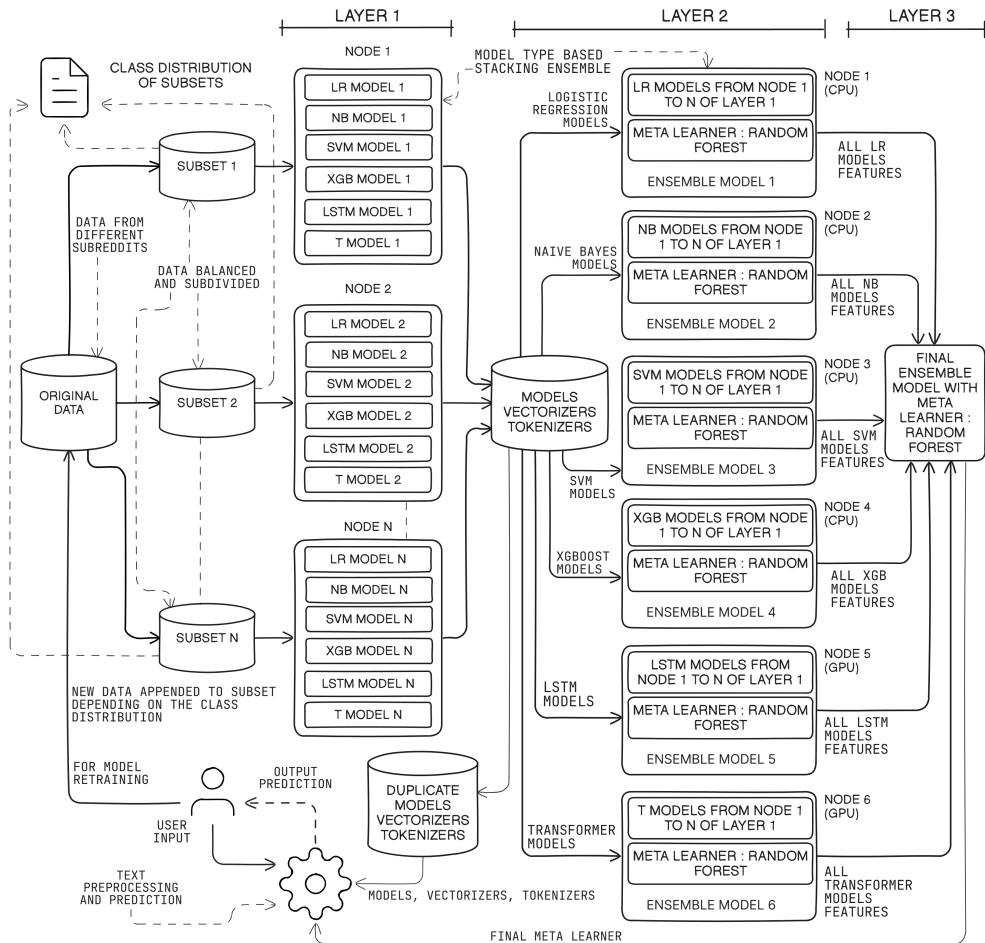
**Note:** The final ensemble models from two different architectures achieved accuracies of **96.24%** and **96.25%** respectively.



Scalable Distributed Architecture 1

#### Summary of Architecture 1: Hierarchical Ensemble

Aspect	Description
Data Partitioning	The dataset is divided into manageable subsets for independent processing.
Base Models	Each subset trains base models (Logistic Regression, Naive Bayes, SVM, LSTM, XGBoost, and Custom Transformers).
Subset Ensemble	For each subset, a Random Forest meta learner is used to combine the base models into a subset-specific ensemble.
Global Aggregation	The subset-specific ensembles are aggregated to form a final global ensemble, capturing comprehensive learning.
Scalability	Designed for distributed processing; subsets can be processed in parallel across multiple nodes.
Fault Tolerance	Modular design ensures that failure in one subset does not impact the overall ensemble performance.
Distributed Processing	Worker nodes independently train base models and report to a central controller that aggregates the results.



Scalable Distributed Architecture 2

### Summary of Architecture 2: Optimized Hierarchical Ensemble

Aspect	Description
Optimized Aggregation	Models of the same type from all subsets are combined into a single intermediate ensemble for that model type.
Intermediate Ensembles	Produces six intermediate ensembles (one each for Logistic Regression, Naive Bayes, SVM, XGBoost, LSTM, and Transformer).
Final Ensemble	The six intermediate ensembles are aggregated via a meta learner (Random Forest) to form the final ensemble.
Fixed Ensemble Count	The number of intermediate ensembles is fixed at six, regardless of the number of subsets, reducing computational overhead.
Efficient Resource Utilization	GPUs are reserved for training the computationally intensive LSTM and Transformer ensembles; lightweight meta learners (e.g., Logistic Regression or Random Forest) optimize aggregation.
Data Bus Implementation	A structured data bus efficiently transfers models from subsets to intermediate nodes, reducing latency and synchronization overhead.
Improved Cross-Validation	Streamlined aggregation minimizes redundancy, resulting in faster cross-validation and training on large datasets.

To determine  $n$ , the number of subsets, based on the size of the original dataset, it is crucial to consider computational efficiency, memory constraints, and sequential execution. Given that the dataset has  $D = 167,229$  records and is processed sequentially in Google Colab with 12GB of RAM per node, the number of subsets  $n$  must strike a balance between memory usage and model performance. In this case, you chose  $n = 6$  subsets, which implies a subset size  $S$  of:

$$S = \frac{D}{n} = \frac{167,229}{6} \approx 27,872 \text{ records per subset.}$$

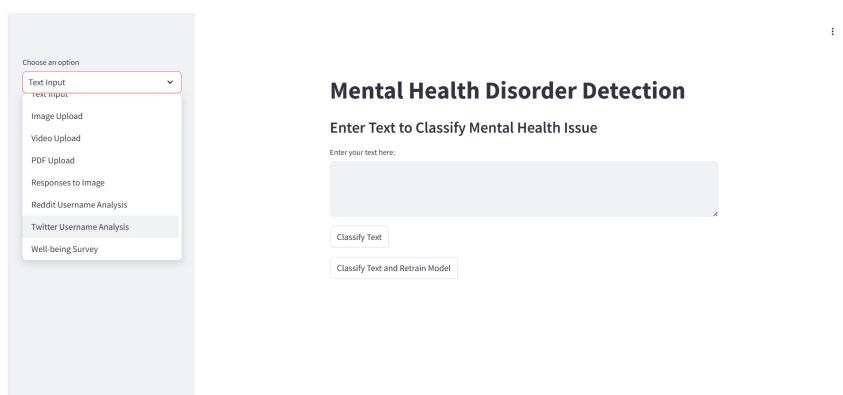
The choice of  $n = 6$  is reasonable given the following factors:

1. **Memory Constraints:** Google Colab provides 12GB of RAM. Each subset must fit within this memory while accommodating the model's requirements for training and validation. Processing approximately 27,833 records at a time is well-suited to this memory limit for most machine learning models, including Logistic Regression, SVM, LSTM, and Transformer-based models.
2. **Sequential Execution:** Since the architectures are implemented sequentially, the number of subsets  $n$  does not need to align with the number of computational nodes. Instead, the goal is to divide the dataset into manageable chunks that reduce training time and memory overhead for each subset.
3. **Performance and Aggregation:** With  $n = 6$ , both architectures remain computationally feasible. In Architecture 1,  $n = 6$  leads to six independent intermediate ensemble models, which are aggregated into the final ensemble. In Architecture 2, models of the same type from all six subsets are combined into six intermediate ensemble models, one for each type of algorithm.

The web application uses only the model from the first intermediate node of Architecture 1, as retraining the larger dataset model takes over 20 minutes.

## 9.5 User Interface of the Application

Below are some snapshots from the web application.



List of Options for the User

# MAFSMBMDDPW1

**Mental Health Disorder Detection**

Enter Text to Classify Mental Health Issue

Enter your text here:

```
অসম বৰ কা হোকোঁ। সাগৰ শৰীরে এক বন্দনা আছিৱৰা অনুভূত হয়, সুন্দৰ যথোচিত চাপ বাঁচোৱা থাকে, খুল মিনোৰ কৰা বৰা ঝুঁটু জামো, এক বেকাৰ মৰণ সংবেদনৰ বৰা, কিন্তু জৰুৰত আছি মৰণৰ দেৱতাৰ প্ৰতিকৰণৰ প্ৰতিকৰণৰ পথত অনুভূত হয়, ফেৰানে কুমি কোনো কৰণ নাই।
```

Classify Text and Retrain Model

Translated Text (to English):

Your mind seems to be stuck in a thousand thoughts and doubts every day. A feeling like a nerve-wracking, kind of pressure that never goes away. An unknown fear works inside you, which says – “Something bad will happen.” Maybe you know, nothing is likely to happen, but deep down in your mind an impossible fear awakens. A kind of restlessness is felt throughout your body, the pressure in your chest increases, it becomes difficult to breathe. You know, this is just a mistake in concentration, but still it seems to spread to every cell of your body. The world feels like a dark hole, where you are alone and

**Mental Health Disorder Detection**

Upload an Image to Extract and Classify Text

Upload an Image

Drag and drop file here  
Limit 200MB per file - JPG, JPEG, PNG, WEBP, BMP, TIFF, TIF

OIP (2).jpg 26.7KB

Classify Text and Retrain Model

**Text Input Option**

**Upload Image Option**

**Mental Health Disorder Detection**

Upload a Video to Extract and Classify Text

Choose a video file

Drag and drop file here  
Limit 200MB per file - MP4, MOV, AVI, MPEG4

smp3.mp4 3.7MB

Classify Text and Retrain Model

**Mental Health Disorder Detection**

Upload a PDF to Extract and Classify Text

Upload a PDF

Drag and drop file here  
Limit 200MB per file - PDF

Handwritten text.pdf 1.5MB

Extracted Pages

I don't know where to begin. Maybe it doesn't even matter. Lately everything I feel like a blur, like I'm moving through life in slow motion while the rest of the world rushes past me. It's exhausting, wishing up every day knowing it's going to be the same as yesterday. the same weight pressing down on my chest, the same thoughts circling in my head, the same emptiness that never seems to go away.

**Upload Video Option**

**PDF Upload Option**

**Mental Health Disorder Detection**

Describe Image and Classify Responses

Look at the image and answer the questions.

**Answer the following questions:**

What do you see in this image?

I see a woman with a distant, haunted expression. Her head seems to be bursting with chaotic, mechanical fragments, almost like intrusive thoughts or painful memories that she cannot escape. The background is dark and fragmented, with shadowy figures that seem distant yet significant, as if representing people from her past or unresolved traumas.

What emotions does this image evoke in you?

It evokes feelings of anxiety, isolation, and distress. The mechanical tangles in her mind feel heavy, symbolizing an overwhelming mental burden. The color palette, with its contrast between dark blues and fiery oranges, makes me feel trapped between the coldness of detachment and the burning intensity of raw trauma.

Does this image remind you of anything from your past?

Yes, it reminds me of moments where my thoughts feel disconnected, where memories of past pain resurface without warning. The figures in the distance mirror the feeling of being surrounded by others yet feeling completely alone. The construction-like elements in the background resemble a city lost in time, much like certain places used to difficult memories.

If this image had a story, what would it be?

The story would be of a woman struggling with the weight of her past. She walks through life carrying memories that refuse to fade, each piece of her fragmented mind representing a moment of fear, loss, or helplessness. The figures in the distance are the ghosts of the life she has lost or

**User response to Image Option**

**Questions related to image**

**Mental Health Disorder Detection**

Enter Reddit Username for Analysis

Enter Reddit username:

flowerpower0601

Analyze

Recent Text Posts:

```
0 : "Taylor Swift's 'Eras' show. What's ACTUALLY going on? What do guys think of this?"  
1 : "Taylor Swift's 'Eras' show. What's ACTUALLY going on? [removed]"  
2 : "Taylor Swift and die Eras Tour"  
Ich habe Letztens dieses Videos gefunden, welches behauptet das die Eras Tour von Taylor Swift überhaupt nicht live ist und das selbst, dass was sich "live" anbietet einfach pre-recorded ist und die Band gar nicht live spielt sondern die ganze Show ein Backing Track ist, der abgespielt wird.
```

**Mental Health Disorder Detection**

Enter Twitter Username for Analysis

Enter Twitter username:

narendramodi

Analyze

Recent Text Posts from Tweets:

```
0 : "Well said. It is good that this truth is coming out, and that too in a way common people can see it."  
1 : "মহান ভাষণশৈলী এবং কাম্য প্রযোজনীয়তা মী লেখ আসতেৰী অৱি কৰে, মহাশূণ্য বিষয়া আপি মনো লোকৰ স্বাত পাবল কোৱা আৰে তে এই ৩৫ বছৰীক লোক, কৰিবি সেৱাৰী আৰু সুলভী কৰন কৰিবো— অসম মুকুটৰ কৰকাৰ লোক। https://t.co/0mrv3Dufkj"  
2 : "
```

**Reddit User Analysis Option**

**Twitter User Analysis Option**

# MAFSMBMDDPW

chest increases, it becomes difficult to breathe. You know, this is just a mistake in concentration, but still it seems to spread to every cell of your body. The world feels like a dark hole, where you are alone and helpless.

The most likely mental health concern from all the text obtained is: anxiety with a probability of 99.95%

**Wellbeing Insight:**

- Autonomy and Anxiety:** Anxiety can severely impair autonomy. The overwhelming fear and worry associated with anxiety can make it difficult for individuals to confidently assert their opinions or make independent decisions, even when they disagree with the majority ("I have confidence in my opinions, even if they are contrary to the general consensus"). The need for external reassurance and validation becomes prominent, hindering self-regulation and leading to dependence on others for decision-making. This dependence contradicts the Ryff scale's definition of autonomy.
- Environmental Mastery and Anxiety:** Anxiety significantly diminishes environmental mastery. The inability to manage overwhelming feelings can lead to avoidance behaviors, preventing individuals from effectively managing daily tasks and opportunities ("In general, I feel I am in charge of the situation which I live"). Anxiety can manifest as procrastination, difficulty focusing, and an inability to anticipate and plan for future events, resulting in a decreased sense of control over one's environment.
- Personal Growth and Anxiety:** Anxiety often impedes personal growth. The constant worry and fear can stifle exploration of new experiences and opportunities ("I think it is important to have new experiences, that challenge how you think about yourself and the world"). Individuals might avoid situations that

## Prediction and Insights

chronic pain. The user's feelings are valid and understood, but hope and improvement are still possible.

Updated Dataset (Last 3 Rows):

text	mental_health_issue
I was born trying to kill myself Came out looking like fucking meganid. Some basic depression	depression
Sometimes your mind reaches extreme heights, you feel a kind of invincible power - i bipolar	bipolar
I can accept rd2 made with rd1 graphics. It might sound weird, but I really love the z normal	normal

Model is being retrained...

Detailed Status

Retrained Model Accuracy: 98.04%

The most frequently detected mental health concern from all the text obtained is: normal appearing in 55.00% of analyzed text.

Mental health issue distribution across posts:

Mental Health Issue	Count
normal	11
depression	4
psd	3
anxiety	2

**Wellbeing Insight:**

It's important to clarify that "normal" isn't a mental health issue. It's a baseline, a descriptor of typical functioning. Therefore, the following paragraphs explore how the typical range of human experience relates to the six factors of Ryff's Psychological Well-being scale. There's no "impact" to reduce or improve; rather, these are observations on how these factors generally appear within the typical spectrum of mental health.

- Autonomy:** Individuals within the "normal" range generally demonstrate a good degree of autonomy. They are capable of making independent decisions and regulating their behavior, even when facing social pressures. They likely agree with statements like "I have confidence in my opinions, even if they are contrary to the general consensus." However, the level of autonomy can vary widely within this

## Social Media User Analysis

Aggregated Emotion Counts Across All Images

Detected Emotions	Frequency
happy	3
sad	5
angry	1
disgust	0
fear	1
surprise	1
neutral	3

Most Frequent Emotion Across All Images or no Images(Default): sad

Elevated sadness suggests potential depression, while the presence of fear might indicate anxiety or PTSD; seeking professional help for evaluation and exploring coping mechanisms like therapy and mindfulness are crucial.

**Model Retraining Result**

NUNCA SOY LUQUERIA  
FUERTE' SIGNIFICABA, HASTA  
QUE SER FUERTE FUE MI  
UNICA OPCION.

barbiefd.tumblr.com

uploaded image

a drawing of a person with a sign on it

Translated Text (to English)

I NEVER KNEW WHAT "BEING

Generate Image Caption

The most frequently detected mental health concern from all the text obtained is: bipolar with a probability of 64.30%

**Wellbeing Insight:**

Knowledge Graph after prediction

Weighted Sum Analysis

**Weighted Sums**

Weighted Sum	Value
0	2.1109
1	3.0648
2	2.0152
3	2.9005
4	2.8856
5	2.874

Row with Highest Weighted Sum

Index: 1  
Row Name: positive relations with others  
Weighted Sum: 3.064796033041324

**Cosine Similarity Analysis**

**Cosine Similarities**

Cosine Similarity	Value
0	0.5806
1	0.7201
2	0.6985
3	0.6861
4	0.6603
5	0.6594

Row with Highest Cosine Similarity

Index: 1  
Row Name: positive relations with others  
Cosine Similarity Score: 0.7200664859362366

**Weighted Sum Analysis**

**Cosine Similarity Analysis**

**Euclidean Distance Analysis**

**Euclidean Distances**

	Euclidean Distance
0	7.0727
1	8.2532
2	5.4968
3	8.2124
4	8.5131
5	8.093

**Row with Smallest Euclidean Distance**

Index: 2  
Row Name: autonomy  
Euclidean Distance: 5.496762612189636

**Specific Parameter Based Insight:**

A person struggling with depression often finds it challenging to maintain positive relationships and cultivate autonomy. However, focusing on these areas within the Ryff scale can significantly contribute to recovery. Here's practical advice, keeping in mind that progress is gradual and setbacks are normal:

- Improving Positive Relations with Others:
  - Start small, don't overwhelm; instead of aiming for large social gatherings, start with small, manageable interactions. A brief phone call with a trusted friend or a short coffee date is a good beginning. Focus on the quality of connection, not quantity.
  - Identify supportive individuals: Recognize and prioritize connections with people who are understanding, patient, and supportive. They can provide a safe space to express feelings without judgment. Limit contact with those who might exacerbate your depression.
  - Communicate honestly (but cautiously): Sharing your struggles with trusted individuals can be incredibly helpful, but do so at your own pace. Start with small disclosures and gauge their reactions. If you feel judged or invalidated, protect your emotional well-being and limit further disclosure.
  - Practice active listening: Focusing on others helps shift attention away from internal struggles. Truly listening to friends and family strengthens bonds and fosters reciprocal support.
  - Engage in shared activities: Participate in activities you enjoy with others, even if it's just a walk in the park or watching a movie. Shared experiences create connection and positive memories.
  - Join a support group: Connecting with others who understand depression can reduce feelings of isolation and provide a sense of community. Online support groups can offer accessibility if in-person meetings are challenging.
  - Accept help: Don't hesitate to ask for help when needed. This could range from practical assistance

## Euclidean Distance Analysis

## Specific Parameter based Insights

**Well-being Survey**



If you are not sure, predict your probable mental issue using any one of the 6 options available on the left before filling.

There is a total of 12 questions : 2 for each of the 6 parameters from Ryff's Scale of Psychological Wellbeing. The Overall Scores are displayed at the end along with the updated Association Matrix.

Questions with (R) are reversed scored.

1 = Strongly Disagree  
2 = Disagree  
3 = Slightly Disagree  
4 = Slightly Agree  
5 = Agree  
6 = Strongly Agree

**Q00. What is Your Predicted Mental Issue?**

(Anxiety, Bipolar, Depression, Normal, PTSD)

**Q01. When I look at the story of my life, I am pleased with how things have turned out.**

(1, 2, 3, 4, 5, 6)

**Q02. In many ways I feel disappointed about my achievements in life. (R)**

## Well-being survey option

## Well-being survey questions

Responses Submitted Successfully!

**Overall Scores (Max: 12 for each parameter) and Interpretation :**

**Self Acceptance: 10 (High Scorer)**

Possesses a positive attitude toward the self; acknowledges and accepts multiple aspects of self, including good and bad qualities; feels positive about past life.

**Positive Relations with Others: 10 (High Scorer)**

Has warm, satisfying, trusting relationships; concerned about the welfare of others; capable of strong empathy, affection, and intimacy.

**Autonomy: 10 (High Scorer)**

Is self-determining and independent; able to resist social pressures; regulates behavior from within and follows personal standards.

**Environmental Mastery: 10 (High Scorer)**

...

Updated Association Matrix successfully.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	i
6	2	5	3	2	3	4	3	4	3	2	3	4	E
7	3	4	1	2	1	4	1	4	6	5	2	3	J
8	5	1	5	1	5	3	5	1	5	2	5	1	F
9	2	5	4	5	1	6	5	2	5	3	6	3	F
10	6	3	6	3	6	3	6	3	6	3	6	3	F

Total Number of Respondents (2025-02-08): 1

Updated Association Matrix successfully.

**Updated Association Matrix:**

parameter	anxiety	bipolar	depression	normal	ptsd
self acceptance	3	2	1	5	2
positive relations with others	3	4	2	5	3
autonomy	2	3	2	5	1
environmental mastery	2	4	1	5	4
purpose in life	4	4	1	5	3
personal growth	3	3	1	5	4

## Well-being survey result

## Updated Association Matrix after responses

**Wellbeing Insight Using RAG :**

Here's a Ryff Scale wellbeing analysis based on your experiences, tailored to bipolar tendencies:

- Autonomy:** Mood swing impacts sense of control. Advice: Practice mood charting; identify triggers; implement coping mechanisms.
- Environmental Mastery:** Fluctuations hinder daily management. Advice: Establish a highly structured routine; utilize reminders & task management apps.
- Personal Growth:** Cycle fosters negative self-perception. Advice: Focus on resilience; document coping successes; practice self-forgiveness.
- Positive Relations:** Tension and loneliness strain connection. Advice: Seek support from bipolar-specific communities; transparently communicate needs.
- Purpose in Life:** Instability leads to feeling directionless. Advice: Explore values independent of mood; set small, achievable goals.
- Self-Acceptance:** Extreme shifts damage self-image. Advice: Practice radical self-compassion; challenge mood-related self-criticism.

**Match Score:**

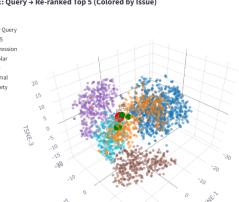
0.4896 (higher is better)

**3D Match Visualization**

3D t-SNE: Query + Re-ranked Top 5 (Colored by issue)

Legend:
 

- Your Query
- Top 5
- Depression
- Bipolar
- Normal
- Anxiety



-- Updating Knowledge Base --

## RAG based Wellbeing Insights

## Match Visualization and Dataset Updation

## 10 Conclusion

Project Summary: Benefits, Future Scope, Distributed Architecture, and Threading Enhancements

Aspect	Summary
<b>Project Benefits</b>	<ul style="list-style-type: none"> <li>• Early detection and intervention through social media analysis.</li> <li>• Leverages ML/DL to process large-scale data for objective, scalable diagnostics.</li> <li>• Reduces mis-/under-diagnosis by providing real-time insights.</li> <li>• Offers reusable components applicable in other domains (e.g., market sentiment, cyberbullying detection).</li> </ul>
<b>Future Scope</b>	<ul style="list-style-type: none"> <li>• Incorporate additional data sources (Facebook, Instagram, niche forums) for comprehensive analysis using bigger datasets.</li> <li>• Integrate real-time monitoring and feedback capabilities.</li> <li>• Enhance user privacy using techniques like differential privacy.</li> <li>• Collaborate with psychologists and ethicists to refine ethical and diagnostic guidelines.</li> </ul>
<b>Distributed Architecture</b>	<ul style="list-style-type: none"> <li>• Partition dataset using Hadoop HDFS, Amazon S3, or Google Cloud Storage.</li> <li>• Use Apache Spark or Dask for parallel processing and training on worker nodes (Docker/VMs).</li> <li>• Train base models (e.g., Logistic Regression, SVM) in parallel and aggregate them via a meta learner (Random Forest, XGBoost).</li> <li>• Aggregate subset-specific ensembles using stacking or weighted averaging, and deploy the final model via TensorFlow Serving, Kubernetes, or AWS SageMaker.</li> </ul>
<b>Threading Enhancements</b>	<ul style="list-style-type: none"> <li>• Implement threading to handle I/O-heavy tasks (e.g., video/image downloading, audio extraction, transcription) concurrently.</li> <li>• Overlap network and disk operations to reduce idle time.</li> <li>• Improve scalability and responsiveness, enabling the system to serve multiple requests simultaneously.</li> </ul>
<b>Custom SLM (Small Language Model) Challenges in Google Colab</b>	<ul style="list-style-type: none"> <li>• Finding proper weights and biases is difficult.</li> <li>• Epoch 1 takes excessively long, making pretraining or fine-tuning on larger datasets unfeasible.</li> <li>• Using pre-trained OpenAI weights builds a foundation but fails to generate meaningful texts.</li> <li>• Generating 1024-token sequences takes approximately 20 minutes, which is impractical.</li> </ul>

## 11 References

- [1] Hatoon S AlSagri and Mourad Ykhlef. Machine learning-based approach for depression detection in twitter using content and activity features. *IEICE Transactions on Information and Systems*, 103(8):1825–1832, 2020.
- [2] Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. Predicting depression via social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 2013.
- [3] Jetli Chung and Jason Teo. Single classifier vs. ensemble machine learning approaches for mental health prediction. *Brain Informatics*, 10(1):1, jan 2023.
- [4] Sharath Chandra Guntuku, David Bryce Yaden, Margaret L. Kern, Lyle H. Ungar, and Johannes C. Eichstaedt. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49, 2017.
- [5] Priya Mathur, Amit Kumar Gupta, and Abhishek Dadhich. Mental health classification on social-media: Systematic review. *Proceedings of the 4th International Conference on Information Management & Machine Intelligence*, 2022.
- [6] Moin Nadeem. Identifying depression on twitter. *arXiv preprint arXiv:1607.07384*, 2016.
- [7] Ramin Safa, S. A. Edalatpanah, and Ali Sorourkhah. Predicting mental health using social media: A roadmap for future development, 2023.
- [8] Dip Kumar Saha, Tuhin Hossain, Mejdl Safran, Sultan Alfarhood, M. F. Mridha, and Dunren Che. Ensemble of hybrid model based technique for early detecting of depression based on svm and neural networks. *Scientific Reports*, 14(1):25470, oct 2024.
- [9] Konda Vaishnavi, U Nikhitha Kamath, B Ashwath Rao, and N V Subba Reddy. Predicting mental health illness using machine learning algorithms. *Journal of Physics: Conference Series*, 2161(1):012021, jan 2022.
- [10] Hongzhi Zhang and M. Omair Shafiq. Survey of transformers and towards ensemble learning using transformers for natural language processing. *Journal of Big Data*, 11(1):25, feb 2024.