Psychological Mental Health Analyzer

(for the partial fulfillment of Bachelor of Technology Degree in Computer Science & Engineering)

Submitted by

Hritik Negi

Kumari Anjali

Saksham Bijalwan

Rishabh Nautiyal

Under the guidance of

Miss. Manika Manwal

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNIVERSITY

2020 - 2024

CERTIFICATE

This is to certify that the project titled "**Psychological Mental Health Analyzer**" submitted by **Hritik Negi, Kumari Anjali, Saksham Bijalwan and Rishabh Nautiyal** to Graphic Era Hill University for the award of the degree of **Bachelor of Technology**, is a Bonafide record of the research work done by them under my supervision. The contents of this project in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Dehradun

Date:

Miss. Manika Manwal

(Asst. Professor)

GEHU, Dehradun

ACKNOWLEDGEMENT

I would like to particularly thank our project guides, Miss. Manika Manwal and Dr. Satvik Vats for their patience, support and encouragement throughout the completion of this project.

At last but not the least I am greatly indebted to all other persons who directly or indirectly helped us during this project.

Kumari Anjali

Roll No. - 2018439

B.Tech CSE -I-VIII SEM

Hritik Negi

Roll No. - 2018384

B.Tech CSE-H-VIII SEM

Rishabh Nautiyal

Roll No.-2018631

B.Tech CSE-H-VIII SEM

Saksham Bijalwan

Roll No. - 2018684

B.Tech CSE - H-VIII SEM

ABSTRACT

This research explores the domain of real-time psychological analysis using pre-stored information and advanced Natural Language Processing (NLP) techniques. The primary aim is to enhance the precision and accuracy of psychological evaluations by analyzing language patterns in live conversations. By leveraging NLP and machine learning techniques, this study seeks to provide a deeper understanding of mental health challenges, with a particular focus on stress prediction among university students, working professionals, and introverted individuals. The research delves into various aspects of mental health, particularly targeting the identification and prediction of stress and depression. By employing sophisticated NLP techniques and machine learning algorithms, the study aims to detect subtle linguistic cues that may indicate psychological distress. The investigation encompasses a broad spectrum of methods for discerning depressive states from textual data, providing a comprehensive comparative analysis of their effectiveness. One of the core contributions of this research is the advancement in understanding live text patterns and their implications for mental health. The study's findings are expected to have significant implications for personalized mental health interventions, enabling more tailored and timely support for individuals experiencing mental health issues. Additionally, the research addresses the ethical considerations associated with the application of these technological advancements in psychological evaluation, ensuring that the deployment of such tool's respects privacy and confidentiality.

TABLE OF CONTENTS

CERTIFICATE **i**

ACKNOWLEDGEMENT **ii**

ABSTRACT **iii**

LIST OF TABLES **iv**

LIST OF FIGURES **v**

ABBREVIATIONS **vi**

NOTATIONS

1. INTRODUCTION

1.1 Project Overview

1.2 Objectives of the Project

1.3 Scope of the Project

2. LITERATURE REVIEW

3. REQUIREMENT ANALYSIS

3.1 Functional Requirements

3.2 Non - Functional Requirements

3.3 Hardware Software Requirements

3.4 User Requirements

3.5 Regulatory and Ethical Requirements

4. PROJECT DESIGN AND IMPLEMENTATION

4.1 Proof of Concept

4.2 Application of Algorithm

4.2.1. Logistic Regression

4.2.2. Naïve Bayes

4.2.3. Support Vector Machine

4.3 Data Collection

4.4 Data Pre-processing

4.4.1. Remove Stop Words

4.4.2. Remove Special Symbol

4.4.3. Remove Non-Alphabetic Character

4.5. Model Training

5. TESTING AND RESULTS OF THE PROJECT

5.1 Result Table

5.2 Result

6. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

6.2 Future Scope

7. PROJECT SNAPSHOTS

APPENDIX

REFERENCES

PUBLICATIONS

LIST OF FIGURES

1. Table 2.1 Major Literature review
2. Fig.3.1.3.1 Prediction Model
3. Fig.3.3.1 dataset distribution
4. Fig.3.3.1. Length of words
5. Fig4.1.1 Methodology
6. Fig.4.2.1 precision of algorithm
7. Fig.4.2. 2.. Performance evaluation
8. Fig. 4.4.1 Steps for data pre-processing
9. Fig.4.6.1. Accuracy on training and validation
10. Fig4.6.2. Losses on training and validation

ABBREVIATION

1. **ML - Machine Learning**
2. **PTSD - posttraumatic stress disorder**
3. **SVM - Support Vector Machine**
4. **AUC - Area Under the ROC Curve**
5. **LSTM - Long Short-Term Memory**
6. **BERT - Bidirectional Encoder Representations from Transformers**
7. **API - Application Programming Interface**
8. **SSD - Solid State Drive**
9. **GPU - Graphics Processing Unit**
10. **IDE - Integrated Development Environment**
11. **OS - Operating System**
12. **RAM - Random Access Memory**
13. **CPU - Central Processing Unit**
14. **PDF - Portable Document Format**
15. **HIPAA - Health Insurance Portability and Accountability Act**
16. **GDPR - General Data Protection Regulation**
17. **CSV - Comma-Separated Values**
18. **JSON - JavaScript Object Notation**
19. **WCAG - Web Content Accessibility Guidelines**
20. **WSD - Word Sense Disambiguation**
21. **PTSD - Post-Traumatic Stress Disorder**
22. **WHO - World Health Organization**

CHAPTER 1

INTRODUCTION

Mental health is a crucial aspect of overall well-being, encompassing emotional, psychological, and social factors that influence how individuals think, feel, and behave. It affects daily functioning, relationships, and the ability to handle stress and adversity. Despite its importance, mental health is often overlooked, leading to significant personal and societal consequences. Globally, millions of people suffer from various mental health disorders, including anxiety, depression, and schizophrenia, which can severely impact their quality of life. The ongoing stigma and lack of adequate resources further exacerbate the challenges faced by those affected. Addressing mental health requires a comprehensive approach, integrating awareness, prevention, and innovative treatment strategies to support individuals in achieving optimal mental health and well-being. According to the WHO, Millions of people all around the world were living with some kind a mental disorder, anxiety and depressive disorders. In 2020, the number of people living with depression and anxiety disorders rise significantly because of the COVID-19 pandemic. Report shows that 26% and 28% increments for anxiety and major depressive disorders respectively in just one year [1]. There is an estimate of people 3.8% suffer from depression, including 5.7% of individuals over the age of 60 and 5% of adults (4% of males and 6% of women). Depression affects over 280 million individuals worldwide. *(2)*. Over 700,000 individuals lose their lives because of suicides every years. Suicide is the 4th leading cause of death in 15- to 29-year-olds [2].

The implication of this research extends the realm idea of personalized mental health support and intervention, driven by real-time assessments, ushering in new possibilities, for enhancing the comprehension and healthcare of an individual psychological well-being. Simultaneously, within the boarder context of mental health, depressive disorder emerges as a severe and pervasive challenge, significantly impacting lives. Given the suicide rank as the 4th leading cause of death globally, there is a critical need for effective diagnostic and intervention strategies. This study addresses these concerns by exploring the potential of machine learning, data mining techniques and specific NLP algorithm to enhance the understating and management of mental health issues. This research project navigates the complexities presented by anxiety and sadness within the intricate landscape of mental health illness. Navel approaches to studying and treating this illness are made feasible through our increasing reliance on computer technology, microelectronics and machine learning. When applied to patient histories, machine learning and data mining techniques, along with specialized NLP algorithms, offer a promising avenue to

improve diagnostic accuracy, facilitating ration decision-making amidst complex and ambiguous data. Furthermore, the research delves into the identification and analysis of depression using textual data through various NLP techniques and machine learning algorithms. The study distinguishes itself by emphasizing a comparative analysis to discern the most effective approaches for detecting depressive states in textual data, making a crucial step forward in understand and addressing mental health challenges through data-driven insight.

The implication of the research transcends academic inquiry holding promising prospects for personalized mental health support and intervention based on real-time assessment. The ability to comprehend and manage individual psychological well-being in real-time present novel opportunities for tailored intervention, potentially revolutionizing mental health care. In the subsequent section, we will delve deeper into the methodologies employed, examine the underplay of NLP algorithm's, data mining and machine learning techniques in real-time psychological analysis. The findings of the research will be presented and analyzed, offering valuable insight into the potential applications of these technologies, particularly NLP algorithm. In advancing our understating and management of mental health issues. The paper will conclude different research and outline the potential direction of future investigating, emphasizing the importance of continued explorations in the critical field of mental health disease on larger scale.

1.1. Project Overview

Mental health disorders, including anxiety, depression, and other psychological conditions, represent a significant global health challenge. Traditional methods for diagnosing and treating these disorders often rely on subjective self-reports and clinical observations, which can be prone to bias and inaccuracies. Advances in Natural Language Processing (NLP) and Machine Learning (ML) offer new avenues for analyzing textual data to provide more objective, real-time insights into individuals' mental health. This project aims to leverage these technologies to develop a robust system for psychological mental health analysis based on text data, enhancing diagnostic accuracy and enabling personalized interventions.

1.2. Objectives of the Project

Objective of the Project

The objective of this project is to utilize machine learning techniques to develop an advanced, real-time system for psychological analysis, with a primary focus on stress and depression detection. Through the integration of NLP algorithms and machine learning models, the project seeks to accurately assess individuals' mental well-being by analyzing language patterns and identifying linguistic markers indicative of stress and depressive states. Ultimately, the goal is to provide timely interventions and support to individuals experiencing psychological distress, contributing to improved mental health outcomes and well-being. The project aims to:

1. Enhance Psychological Evaluation

Enhancing psychological evaluation through Natural Language Processing (NLP) offers valuable insights into individuals' mental well-being. By analyzing language patterns in real-time conversations, NLP techniques can provide immediate feedback on emotional states and cognitive processes, enabling more accurate and timely psychological assessments. Additionally, by identifying linguistic markers indicative of stress and depressive states, these techniques enhance the precision of assessments, facilitating early intervention and support. NLP algorithms can detect subtle nuances in language, such as changes in tone, sentiment, and word choice, which may signify underlying psychological distress. By integrating NLP into psychological evaluation processes, mental health professionals can gain a deeper understanding of individuals' emotional and psychological states, leading to more effective interventions and treatment strategies. Ultimately, this approach contributes to improved mental health outcomes and enhances the quality of care for individuals experiencing stress and depression.

2. Stress and Depression Prediction:

Developing and implementing machine learning models for stress and depression prediction is crucial for early intervention and support across diverse populations. By leveraging data from different groups, including university students, workers, and introverted individuals, these models can identify patterns and risk factors associated with stress and depressive symptoms. Using sentiment analysis techniques, emotional states can be detected from social

media posts and other textual data sources, providing valuable insights into individuals' mental well-being. These models enable proactive interventions and personalized support tailored to individuals' needs, promoting mental health awareness and resilience. Additionally, by targeting specific demographics and populations, such as university students or introverted individuals, these models can address unique challenges and risk factors associated with stress and depression, ultimately contributing to improved mental health outcomes and well-being.

3. Comparative Analysis of Techniques

Conducting a comparative analysis of techniques is vital to discern the most effective approaches for detecting depression from textual data. By evaluating various Natural Language Processing (NLP) techniques and machine learning algorithms, researchers can assess their effectiveness in accurately identifying depressive states. Techniques such as sentiment analysis, topic modeling, and linguistic feature extraction can be compared alongside machine learning algorithms like Support Vector Machines (SVM), Naive Bayes, and deep learning models such as Long Short-Term Memory (LSTM) networks. Through rigorous evaluation and comparison, researchers can determine which combination of techniques and algorithms yields the highest accuracy and reliability in detecting depression. This comparative analysis not only enhances understanding of the strengths and limitations of different approaches but also guides the selection of the most effective models for real-world applications in mental health assessment and intervention.

4. Real-Time Assessment and Intervention:

The development of a system capable of real-time psychological assessments is crucial for timely intervention and support in mental health care. By leveraging advanced machine learning algorithms, such a system can analyze live text data, including chat logs or social media posts, to assess individuals' mental health states accurately. This real-time assessment enables prompt identification of potential issues or crises, allowing for timely intervention and support. Furthermore, the system can facilitate personalized mental health interventions based on the live text analysis, providing tailored recommendations, resources, or support services to individuals in need. By offering timely and personalized interventions, this system contributes to improving mental health outcomes and reducing the risk of escalation in crises. However, it is essential to prioritize user privacy and ethical considerations in the development and deployment of such systems, ensuring that interventions are delivered with sensitivity and respect for individuals' autonomy and confidentiality.

5. Ethical Considerations and Data Privacy:

Ethical considerations and data privacy are paramount in the development and deployment of systems for mental health evaluation. It is imperative to ensure that the developed system adheres to ethical standards, particularly concerning user privacy and data protection. This involves implementing robust data encryption, access controls, and anonymization techniques to safeguard sensitive user information. Additionally, addressing potential ethical implications of using automated systems for psychological evaluation is essential. This includes transparency about how the system operates, ensuring accountability in decision-making processes, and providing users with control over their data. Furthermore, ongoing ethical reviews and consultations with experts in psychology and ethics can help identify and mitigate potential risks and biases associated with automated evaluations. By prioritizing ethical considerations and data privacy, developers can build trust with users and stakeholders while promoting responsible and ethical use of technology in mental health care.

6. Contribution to Mental Health Care:

The utilization of machine learning models in mental health care offers significant contributions to improving patient outcomes and advancing the field. By offering insights and personalized tools, these models enable tailored mental health support, empowering individuals to manage their well-being effectively. Moreover, data-driven methodologies enhance the understanding of mental health issues, facilitating more informed decision-making and targeted interventions. Healthcare professionals' benefit from reliable tools for early detection and intervention in mental health disorders, leading to timely and effective treatments. Ultimately, the integration of machine learning in mental health care not only enhances patient care but also fosters innovation and collaboration in addressing complex challenges in mental health management.

1.3. Scope of the Project

The project, titled "Psychological Mental Health Analysis" focuses on analyzing mental health issues using Natural Language Processing (NLP) and machine learning techniques. The project will investigate and explore the application of NLP and machine learning models within the scope of mental health problems.

Mental health problems will be categorized into five types: schizophrenia, anxiety and depression, bipolar disorder, posttraumatic stress disorder (PTSD),

and mental health problems among children. The data for mental health problems will be collected through various domains and sources.

The project will review and highlight the implementation of NLP and machine learning models in each mental health problem. The machine learning approaches will be divided into supervised learning, unsupervised learning, ensemble learning, neural networks, and deep learning. The machine learning models will be classified based on the type of learning approaches.

The performances of the machine learning models will be included in the project to show the efficiency of the NLP and machine learning approaches within the mental health field. The performances such as accuracy, the area under the ROC curve (AUC), F1-score, sensitivity, or specificity will be specified and mentioned in the project to provide further analysis.

CHAPTER 2

LITERATURE REVIEW

The landscape of mental and emotional well-being assessment is evolving with the rise of social media. Traditional methods like face-to-face interview and self-report questionnaires are reactive and time consuming. The emergence of platforms like Reddit, Twitter and Facebook has opened up new avenues for real time proactive monitoring. People willingly share their daily experiences and emotions online providing a wealth of data for healthcare and wellness professionals. Analysis social media activity allows for a more dynamic and timely understanding of individual's mental state fostering proactive interventions and personalized support in the ever-changing landscape of mental health and emotional well-being.

As the study conducted by Reshma Radheshamjeev Baheti[3], she developed a method for detecting stress and enjoyment using tweeter datasets. The techniques of collecting necessary emotion-related textual information are known as sentimental analysis. They used the framework tensi Strength for identifying the sentiment strength on online communities. Using the Mgram with SVM (Support Vector Machine) 67% of Precision and recall is achieved, for reorganization strength. Both neutral and conventional, as well as an analysis of the data's complexity and diversity in each category. To rank the emotion, machine learning algorithms like as the K-nearest neighbor (KNN) and SVM (support vector machine) classifiers are applied. In another study conducted by Disha Sharma [4] tries to low the pitfall of the stress for understudy's students by examine the effectives of ML (machine learning) algorithms.

TensiStrength is an automated system designed to detect the strength of stress and relaxation expressed in short text messages with a particular focus on transportation related stress. The system employs a lexical approach and linguistic rules to identify stress indicator approach and linguistic rules to identify stress indicator in social media text message contributing to application such as Intelligent Transportation Systems. TensiStrength is evaluated against a sentiment analysis program and generic machine learning algorithms showcasing its effectiveness in stress and relaxation detection in tweets. The

result indicates that TensiStrength performs slightly better than the sentiment analysis program demonstrating its partiality for application requiring stress information. However generic machine learning methods outperform both in overall accuracy the study emphasizes the importance of considering the natural of text analysis and the task purpose when choosing between TensiStrength and generic machine learning approach. The research aims to enhance the predictive power of system like (ITS) by incorporating stress information derive from social media text [5]

This research paper explores the application of Natural Language Processing (NLP) and sentiment analysis to detect mental health issues through the analysis of online social media text. The paper emphasizes the significance of mental health, especially during a pandemic, and highlights the role of NLP in analyzing sentiments on platforms like Twitter to assess individual's psychological well-being. The study aims to detect depression among social media users by analyzing their tweets, providing valuable insights for

healthcare professionals and forensic experts. The model that introduces involves gathering data from online platforms (social media). The next step is preprocessing of data and utilizing sentimental analysis to categorize the post onto neutral, negative and positive emotions. The module gives a rating based on the analysis of sentiment and allowing healthcare professionals to examine early in clarifying mental health issues. Additionally, this paper tells about emotions detection system by machine learning or lexicon, this emphasis the potential tones in the content of online platform. The proposed module performs data mining techniques to predict various aspects of a user's mental state while prioritizing user privacy. Overall, the research aims to contribute to early detection and intervention in mental health through innovative technological approaches [6].

SentiStrength, a novel algorithm designed in informal English text, particularly focusing on positive and negative emotions. Addressing the challenges of identifying sentiment in short, informal messages on platforms like Myspace, the paper highlights the limitation of exiting sentiment detection algorithms, emphasizing the need for a specialized approach. SentiStrength utilized machine learning to optimize sentiment term weightings and incorporates methods for extracting sentiment form non- standard spelling, contributing to its efficacy in predicting positive sentiment strength with 60% accuracy and negative sentiment with 72% accuracy in space comments. The paper compares

SentiStrength's performance with various machine learning algorithm's, demonstrating its superiority, especially in positive sentiment detection. The study acknowledges the difficulty of detecting negative sentiment in informal text due to language creativity and the dominance of positive expression. The finding suggests potential application in automatically identifying positive sentiment in web communication environment and commercial sentiment analysis [7].

Table 2.1 Major Literature review

PUBLICATION	INPUT	METHOD
Reshma Radheshamjeev Baheti [3]	Twitter datasets	Sentiment analysis using TensiStrength, Mgram with SVM, K-nearest neighbor (KNN) classifiers
Disha Sharma [4]	Reedit and twitter	Machine learning algorithms to examine the effectiveness in reducing stress
Umar Rrahid [3]	Textual decision, Social media text from platforms like Twitter	TensiStrength for stress detection, comparison with sentiment analysis programs and generic machine learning algorithms
Zhento Xu [9]	Flickr, Informal English text from Myspace	SentiStrength algorithm, machine learning to optimize sentiment term weightings, comparison with other machine learning algorithms

CHAPTER 3

REQUIREMENT ANALYSIS

In designing a system for Psychological Mental Health Analysis using NLP and Machine Learning, both functional and non-functional requirements are crucial for ensuring effectiveness, efficiency, and reliability. Functional requirements specify the system's behavior and capabilities, such as data collection, analysis, and reporting. Non-functional requirements define attributes like performance, security, and usability, ensuring the system meets quality standards. By delineating these requirements, the system can effectively address mental health challenges, provide timely interventions, and maintain user trust. This holistic approach ensures that the solution not only functions as intended but also meets user expectations and regulatory standards.

3.1 Functional Requirements

1. Data Collection

The data collection phase of the system involves gathering diverse textual data from social media platforms like Twitter, Reddit, and Facebook, focusing on posts, comments, and conversations related to mental health, emotions, and well-being. To ensure effective training and validation of machine learning models, the dataset should contain a substantial number of instances. This is achieved by developing scripts or APIs to retrieve data from social media platforms programmatically. Additionally, web scraping techniques are utilized to extract relevant textual content from public forums and discussion threads.

Robust data storage mechanisms are implemented to organize and manage the collected data efficiently, ensuring data integrity and security. This involves structuring the data in a way that facilitates easy access and retrieval while maintaining its accuracy and consistency. Various storage solutions, such as relational databases or NoSQL databases, may be employed based on the nature and volume of the data.

Moreover, data preprocessing steps may be applied to clean and standardize the collected textual data, removing noise and irrelevant information while retaining essential content for analysis. Techniques like text normalization, tokenization, and stop-word removal may be utilized to prepare the data for further analysis and modeling.

Throughout the data collection process, privacy and ethical considerations are paramount. Steps are taken to anonymize and de-identify sensitive information, ensuring compliance with data protection regulations and safeguarding the privacy of individuals contributing to the dataset. By adhering to ethical guidelines and implementing robust data collection practices, the system can build a comprehensive and reliable dataset for training and validating machine learning models aimed at improving mental health analysis and support.

2. Data Processing

In the data processing phase, the system cleanses the collected data by removing non-alphabetic characters, special symbols, and irrelevant information to enhance its clarity and quality. This involves tokenizing sentences to break them down into individual words or tokens, facilitating further analysis. Additionally, the removal of stop words and the application of other preprocessing techniques are essential steps in preparing the data for analysis.

Preprocessing algorithms are developed using libraries like NLTK (Natural Language Toolkit) or SpaCy to clean and tokenize textual data effectively. These libraries offer a wide range of functions and tools for text processing, including tokenization, stop word removal, and lemmatization.

Regular expressions and string manipulation functions are utilized to remove non-alphabetic characters and special symbols from the text, ensuring that only relevant information is retained for analysis. Implementing data pipelines is crucial to automate the preprocessing workflow, ensuring consistency and reproducibility of results. These pipelines orchestrate the sequence of preprocessing steps, from data cleansing to tokenization, and can be easily adjusted or scaled as needed.

By incorporating preprocessing algorithms and data pipelines, the system streamlines the data processing workflow, enabling efficient preparation of textual data for analysis. This ensures that the subsequent stages of the analysis, such as feature extraction and model training, are based on high-quality and standardized data, leading to more accurate and reliable results. Additionally, automation of the preprocessing workflow reduces manual effort and minimizes the risk of errors, enhancing the overall efficiency and effectiveness of the system.

3. Machine Learning Model

In developing a machine learning model for mental health state prediction, the selection of appropriate algorithms is pivotal. Models like Support Vector Machines (SVM), Naive Bayes, logistic regression, and deep learning models such as Long Short-Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT) offer varying degrees of complexity and performance characteristics, catering to different data types and tasks. These models are trained on preprocessed datasets to learn patterns associated with mental health states. Evaluation of trained models using metrics like accuracy, precision, recall, and F1 score is crucial to assess their performance accurately.

To facilitate model development and evaluation, scripts or notebooks are developed utilizing machine learning frameworks like Tensor Flow, Py Torch, or Scikit-learn. These tools provide a rich set of functionalities for building, training, and evaluating machine learning models efficiently. Experimentation with different algorithms and hyper parameters is essential to optimize model performance, considering factors such as over fitting, under fitting, and model complexity. Techniques like cross-validation and hyper parameter tuning are implemented to ensure the robustness and generalization of the models. Cross-validation helps validate model performance across different subsets of data, while hyper-parameter tuning fine-tunes model parameters to achieve optimal performance on unseen data.

Iterative refinement of models based on evaluation results and domain expertise is crucial to ensure the model's effectiveness in real-world applications. Furthermore, documentation of the model development process, including data preprocessing steps, algorithm selection rationale, and evaluation metrics, is essential for reproducibility and transparency. By following best practices in model development and evaluation, stakeholders can have confidence in the reliability and efficacy of the deployed machine learning model for mental health state prediction.

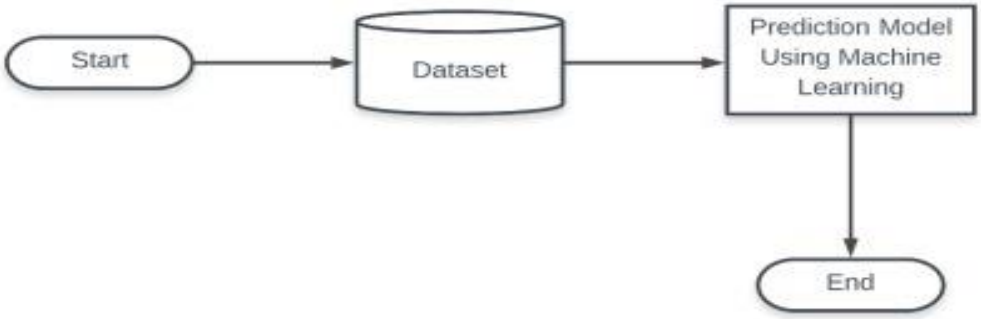


Fig.3.1.3.1 Prediction Model

3.2. Non-Functional Requirements

1. Accuracy

Ensuring accuracy is paramount in predicting mental health states from textual data. The system must achieve high accuracy to deliver reliable results, thereby supporting effective interventions and support. Evaluation of model performance using appropriate metrics and benchmarks is essential to validate the accuracy of predictions rigorously. This involves assessing metrics such as precision, recall, and F1 score, as well as comparing model performance against established benchmarks. By prioritizing accuracy and conducting thorough evaluations, the system can provide trustworthy insights into individuals' mental well-being, ultimately contributing to improved mental health outcomes and well-being.

2. Speed

The project aims to enhance speed by developing efficient algorithms and data processing pipelines to minimize latency in analyzing textual data. By optimizing model inference time, real-time analysis and decision-making, especially in critical scenarios, are enabled. This involves streamlining data processing workflows and implementing techniques to reduce computational overhead, ensuring timely insights into individuals' mental well-being. Ultimately, the goal is to provide swift interventions and support to individuals

experiencing stress and depression, maximizing the system's effectiveness in improving mental health outcomes.

3. Portability

The project emphasizes portability by designing the system to deploy seamlessly across diverse environments and platforms, including local machines, cloud infrastructure, and edge devices. Compatibility with various operating systems, programming languages, and cloud services is ensured, facilitating smooth deployment and integration. This approach enables the system to adapt to different deployment scenarios, maximizing accessibility and usability across a wide range of environments. By prioritizing portability, the project enhances flexibility and scalability, empowering users to leverage the system's capabilities effectively, regardless of their infrastructure preferences or constraints.

4. Scalability

Scalability is integral, with the system designed to accommodate growing data volumes and user demand over time. Implementing distributed computing techniques and cloud-based infrastructure ensures efficient handling of large-scale data processing and analysis. By architecting for scalability, the system can dynamically scale resources to meet evolving demands, maintaining optimal performance and responsiveness. This approach enables seamless expansion to handle increased workloads and ensures continued reliability and performance, even as data volumes and user interactions grow. Ultimately, prioritizing scalability ensures the system's ability to adapt and thrive in dynamic environments, supporting its long-term viability and effectiveness.

5. Reliability

Reliability is paramount, with the system engineered to operate seamlessly under diverse conditions and workloads, mitigating the risk of downtime or failure. Error handling mechanisms are implemented to detect and gracefully handle exceptions, ensuring uninterrupted operation. Thorough testing and validation procedures are conducted to verify the system's reliability and robustness, addressing potential issues proactively. By prioritizing reliability, the system instills confidence in users, guaranteeing consistent performance and availability, even in challenging environments. This approach ensures that individuals relying on the system for mental health support can access

assistance reliably whenever needed, promoting trust and effectiveness in the system's operation.

3.3 Hardware And Software Requirements

I. Hardware Requirements:

Development Environment

1. **Processor:**
    - A multi-core processor is essential for handling parallel processing tasks, which are common in machine learning and data preprocessing.
    - 8 cores or more to ensure efficient processing of large datasets and model training tasks.
2. **Memory (RAM):**
    - Sufficient RAM is necessary to manage the in-memory operations of data preprocessing, feature engineering, and model training.
    - **:** 16 GB to handle moderate data sizes and initial model development.
3. **Storage:**
    - Fast storage solutions such as SSDs are crucial for quick data access and efficient read/write operations.
    - 500 GB SSD to store datasets, intermediate processing files, and model artifacts.
4. **Graphics Processing Unit (GPU):**
    - A dedicated GPU significantly accelerates deep learning model training and inference, especially for neural network architectures.
    - NVIDIA GTX 1080 Ti or RTX 2080.
      .

2. Production Environment

1. **Processor:**
    - Server-grade processors provide the necessary computational power and reliability for handling multiple concurrent requests in a production environment.
    - 4 cores or more to efficiently manage high loads and maintain system responsiveness.
2. **Memory (RAM):**
    - Higher RAM capacity ensures that the system can handle multiple simultaneous inferences and data processing tasks without performance degradation.

- 32 GB for basic production environments.

3. **Storage:**
    - Large and fast storage solutions are essential for maintaining processed data, model versions, and logs.
    - 1 TB SSD for basic data and model storage.
4. **Graphics Processing Unit (GPU):**
    - High-end server-grade GPUs are required for efficient model inference, especially in real-time applications.
    - Multiple GPUs for load balancing and redundancy, ensuring that the system remains responsive under high demand.

5. **Networking:**
    - High-speed internet connectivity is crucial for data collection, real-time API interactions, and cloud-based operations.
    - Reliable broadband connection.

II. Software Requirements:

1. Operating System: An Operating System (OS) is fundamental software that manages computer hardware and software resources, providing services for computer programs. It acts as an intermediary between users and the computer hardware. Key functions of an OS include managing memory, processing tasks, handling input/output operations, and ensuring security and access control. Popular operating systems include Windows, macOS, Linux, and Unix. They offer a user interface, typically graphical, to interact with the system, run applications, and manage files. The OS ensures efficient and secure operation of the computer, enabling users and applications to perform tasks effectively and reliably

- **Development Environment:**
    The operating system should support development tools and libraries used for machine learning and data analysis.

    - Windows 10/11, macOS, or a Linux distribution (e.g., Ubuntu 20.04 LTS).
- **Production Environment:**
    - A stable and secure operating system is essential for reliable production deployments.
    - Linux distribution such as Ubuntu Server 20.04 LTS, CentOS, or Red Hat Enterprise Linux.

2. Programming Languages

- **Python:** Python is a high-level, interpreted programming language known for its simplicity and readability, making it ideal for both beginners and experienced developers. Created by Guido van Rossum and first released in 1991, Python emphasizes code readability and significant indentation, enhancing clarity and organization. Its versatility supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python's extensive standard library and vast ecosystem of third-party packages, available through PyPI, facilitate a wide range of tasks from web development to data analysis and scientific computing.

  Python's syntax is straightforward, closely mirroring the English language, which reduces the learning curve and boosts productivity. It uses indentation to define code blocks, promoting a clean, consistent style. Python is cross-platform compatible, running seamlessly on Windows, macOS, and Linux. Its popularity in data science and machine learning is notable, with powerful libraries like NumPy, pandas, and scikit-learn, as well as web development frameworks like Django and Flask, and tools like Jupyter Notebook for interactive exploration.

Libraries of python use in this project:

- **Pandas:** is a powerful and popular open-source data manipulation and analysis library for Python. Developed by Wes McKinney in 2008, it provides data structures and functions needed to work seamlessly with structured data. The core data structures in pandas are the Data Frame and Series, which facilitate efficient data manipulation and analysis. Pandas excel in handling diverse data formats, including CSV, Excel, SQL databases, and JSON. It offers robust data cleaning, aggregation, and transformation capabilities, making it indispensable for data wrangling tasks. Users can easily perform operations like filtering, grouping, merging, and reshaping data. The library integrates well with other data science tools and libraries, such as NumPy, Matplotlib, and scikit-learn, enhancing its utility in data analysis workflows. Pandas' intuitive API and comprehensive documentation make it accessible to both beginners and experienced practitioners, solidifying its role as a cornerstone of the Python data science ecosystem.

- **NumPy:** NumPy, short for Numerical Python, is an essential open-source library for numerical and scientific computing in Python. Created by Travis Oliphant in 2005, it provides support for large, multi-dimensional arrays (Nd arrays) and a suite of mathematical functions to operate on these arrays efficiently. NumPy's optimized performance, achieved through C and FORTRAN code, allows for fast element-wise operations, linear algebra, and more. It serves as the backbone for many data science libraries, including pandas and Tensor Flow, making it indispensable for tasks in data analysis, machine learning, and scientific research.

- **Matplotlib:** Matplotlib is a widely used open-source plotting library for Python, created by John D. Hunter in 2003. It provides comprehensive tools for creating static, interactive, and animated visualizations in Python. With its versatile API, Matplotlib can generate a variety of plots, such as line graphs, bar charts, histograms, and scatter plots, making it indispensable for data visualization. It integrates seamlessly with other scientific libraries like NumPy and pandas, enhancing its utility in data analysis workflows. Matplotlib's flexibility and extensive customization options make it a powerful tool for visualizing complex data and presenting results effectively.

- **Scikit-learn:** Scikit-learn, commonly known as sklearn, is a powerful open-source machine learning library for Python. It provides simple and efficient tools for data mining and data analysis, built on top of NumPy, SciPy, and Matplotlib. Developed by David Cournapeau in 2007, sklearn offers a wide range of machine learning algorithms, including classification, regression, clustering, and dimensionality reduction. The library's design emphasizes ease of use, consistency, and performance, making it accessible for both beginners and experts. Its comprehensive suite of tools for model selection, preprocessing, and evaluation allows users to build and validate predictive models effectively. Sklearn's robust documentation and active community support further enhance its utility in machine learning projects.

- **NLTK**: The Natural Language Toolkit (nltk) is a comprehensive open-source library for natural language processing (NLP) in Python. Developed initially by Steven Bird and Edward Loper in 2001, nltk provides easy-to-use interfaces to over 50 corpora and lexical resources, such as WordNet. It includes a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Nltk is widely used for educational purposes and research due to its simple and intuitive API. It supports various NLP tasks, such as tokenization (breaking text into words or sentences), part-of-speech tagging, named entity recognition, and text classification. The library also offers tools for cleaning and pre-processing text, which are essential steps in any NLP pipeline. With its extensive documentation and active community, nltk is a valuable resource for developers and researchers working on text analysis, language modeling, and other NLP applications.

3. Development Tools

- **Integrated Development Environment (IDE):** An Integrated Development Environment (IDE) is a software application that provides comprehensive facilities to programmers for software development. It typically includes a code editor, debugger, and build automation tools. IDEs offer features like syntax highlighting, code completion, and version control integration, enhancing productivity and code quality. Popular IDEs, such as Visual Studio Code, PyCharm, and Eclipse, support multiple programming languages and frameworks. They simplify the development process by providing a unified interface and powerful tools, making it easier to write, test, and debug code efficiently. IDEs are essential for both beginners and experienced developers, streamlining the software development lifecycle

IDE Used: PyCharm, Visual Studio Code, or Jupyter Notebook for a comprehensive development experience.

4. Version Control

Version control is a system that manages changes to files over time, enabling multiple users to collaborate on a project. It tracks modifications, allowing users to revert to previous versions, compare changes, and understand the history of a project. Popular version control systems include Git, Subversion (SVN), and Mercurial. They facilitate branching and merging, enabling developers to work on separate features simultaneously without conflicts. Version control is essential for maintaining code integrity, enabling teamwork, and managing large codebases. Tools like GitHub and GitLab enhance these systems by providing cloud-based repositories and additional collaboration features.

Tools Used: Git, with repositories hosted on GitHub, GitLab, or Bit bucket.

5. Dataset

The dataset downloaded from Kaggle titled "Depression" offers a rich source of information for exploring various aspects of depression. This dataset typically includes diverse variables such as demographic information, mental health history, symptoms, and possibly responses to different treatment modalities. Researchers and data analysts can leverage this dataset to perform a multitude of analyses, ranging from descriptive statistics to complex predictive modeling.

By examining patterns and correlations within the data, one can identify risk factors, common symptomatology, and potentially effective interventions. Additionally, machine learning algorithms can be trained to predict the onset or severity of depression based on identified predictors. This dataset is invaluable for advancing our understanding of depression, improving diagnostic tools, and tailoring personalized treatment plans. Its comprehensive nature allows for the development of models that can enhance mental health services and outcomes, providing crucial insights for clinicians, policymakers, and researchers.

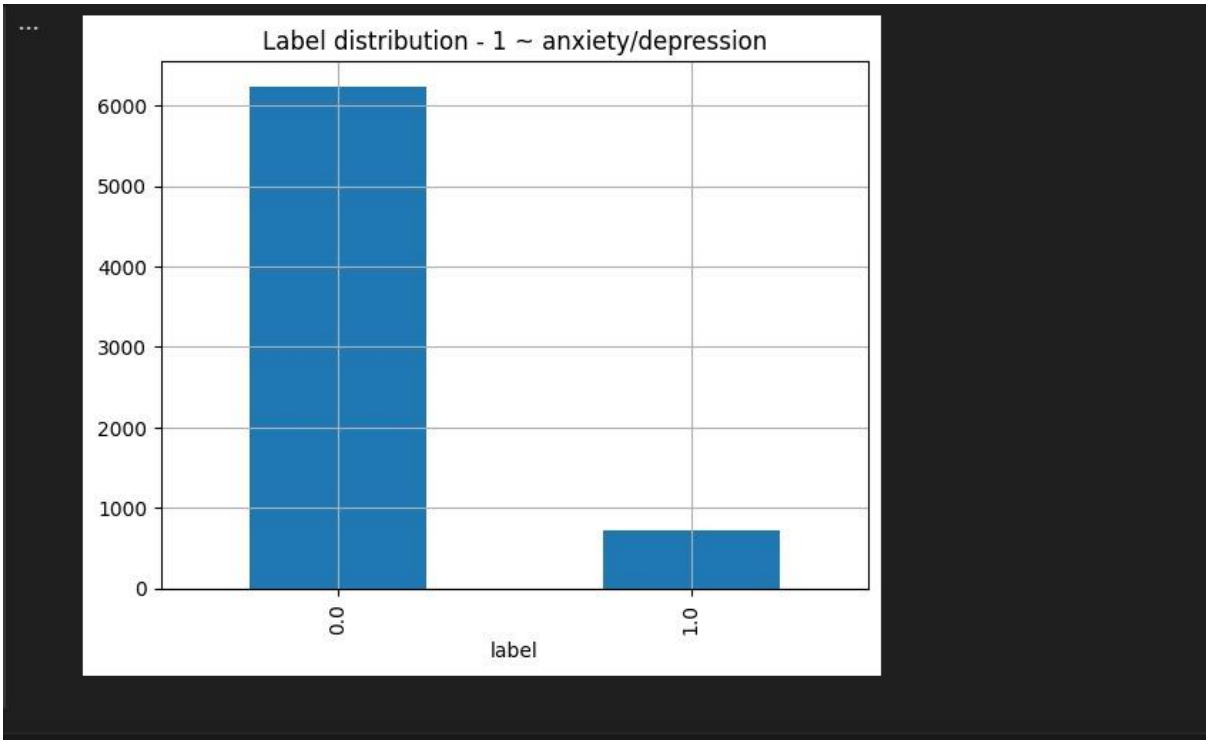


Fig.3.3.1 dataset distribution

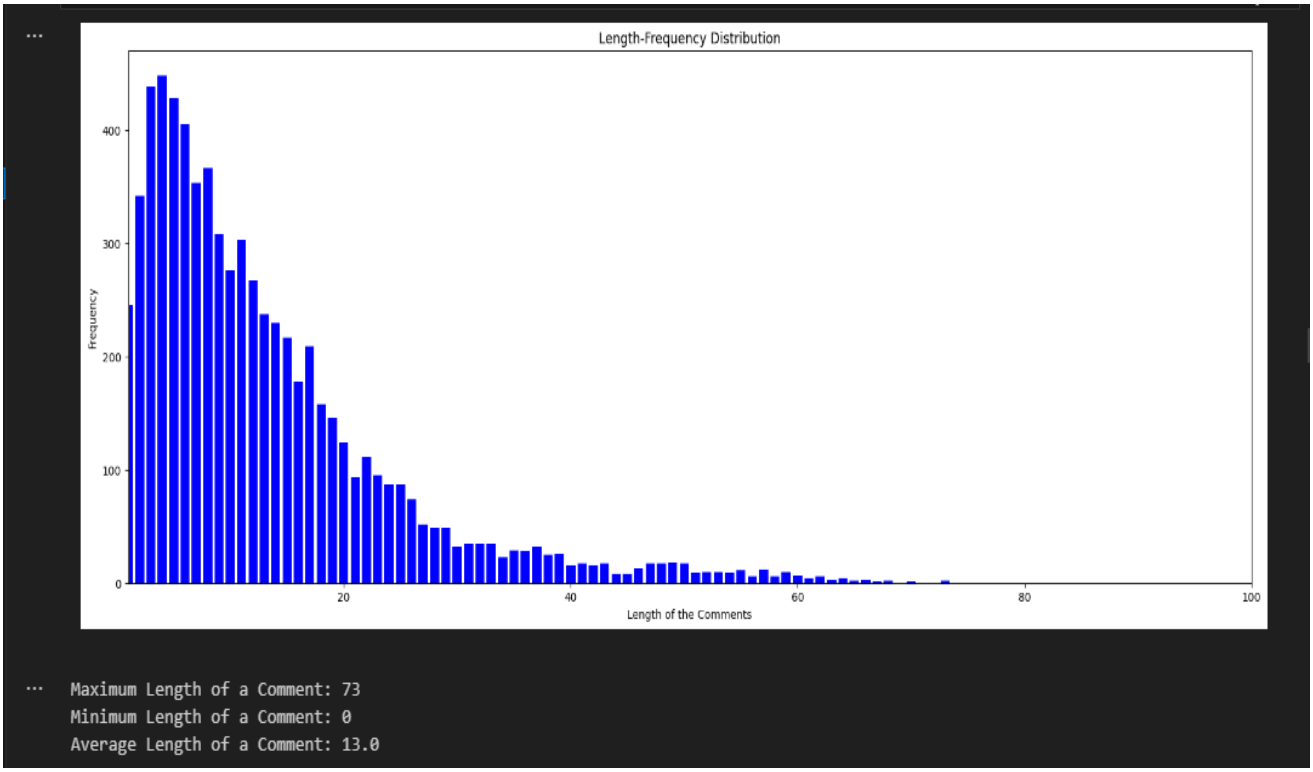


Fig.3.3.1. Length of words

6. Module

A module is a distinct, self-contained unit within a larger system that performs a specific function and can be independently developed, tested, and maintained. In software development, a module refers to a file or collection of files containing code, such as functions, classes, or variables, that can be reused across different parts of a program. Modules enhance modularity, enabling easier debugging, scalability, and collaboration. In education, a module is a segment of a course focusing on a particular topic or skill. Overall, modules contribute to the efficient organization and management of complex systems.

Module used:

- **LOGISTIC_MODEL**

A logistic model, often referred to as logistic regression, is a statistical method used for binary classification tasks. Unlike linear regression, which predicts continuous outcomes, logistic regression predicts the probability of a binary outcome (e.g., success/failure, yes/no). The model uses the logistic function (sigmoid function) to map predicted values to probabilities between 0 and 1.

The logistic function is defined as:

$$ P(Y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1X_1 + \beta_2X_2 + ... + \beta_nX_n)}} $$

where $P(Y=1)$ is the probability of the event occurring, β_0 is the intercept, $\beta_1, \beta_2, ..., \beta_n$ are the coefficients for the predictor variables $X_1, X_2, ..., X_n$, and e is the base of the natural logarithm. Logistic regression is widely used in fields such as medicine, finance, and social sciences for tasks like disease prediction, credit scoring, and customer classification. It is valued for its simplicity, interpretability, and effectiveness in handling binary outcomes.

- **SUICIDE_LOGISTIC_MODEL**A suicide logistic model is a type of logistic regression used to predict the probability of suicide risk based on various predictor variables. This model helps identify individuals at high risk by analyzing factors such as demographic information, mental health history, social support levels, past suicidal behavior, substance abuse, and other relevant psychological and environmental variables.

  The logistic function used in the model outputs probabilities between 0 and 1, indicating the likelihood of an individual attempting or committing suicide. The formula is: $$ P(Y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1X_1 + \beta_2X_2 + ... + \beta_nX_n)}} $$ where $P(Y=1)$ represents the probability of suicide, β_0 is the intercept, and $\beta_1, \beta_2, ..., \beta_n$ are the coefficients for the predictor variables $X_1, X_2, ..., X_n$.

  Such models are crucial for mental health professionals as they provide a data-driven approach to early intervention and prevention strategies, ultimately aiming to reduce suicide rates through targeted support and resources.

- **SUICIDE_XGB_MODEL**
  A suicide XGBoost (extreme Gradient Boosting) model is a powerful machine learning approach used to predict suicide risk by leveraging the XGBoost algorithm. This model is particularly effective for handling large, complex datasets with many features, providing robust predictions based on various risk factors such as demographic details, mental health history, behavioral data, social factors, and clinical information.

  XGBoost builds an ensemble of decision trees in a sequential manner, optimizing for accuracy by minimizing a specified loss function and applying regularization techniques to prevent overfitting. The model's

prediction for an individual's suicide risk is obtained by aggregating the outputs of multiple decision trees.

The formula for the XGBoost model is: [y_i = \sum_{k=1}^K f_k(x_i) \]where \(y_i \) is the predicted risk, \(x_i \) are the input features, \(f_k \) are the individual decision trees, and \ (K \) is the total number of trees.

Suicide XGBoost models are valued for their high predictive performance and ability to uncover complex patterns in data, aiding mental health professionals in identifying high-risk individuals and implementing timely interventions.

3.4. User Requirements:

The user requirements for the system encompass a wide range of functionalities and performance criteria. Users seek real-time psychological analysis capabilities, stress prediction, and depression detection, particularly among university students, workers, and introverted individuals. Accuracy, efficiency, and usability are key priorities, with a focus on providing clear and actionable insights through a user-friendly interface. Security and data privacy are critical, demanding stringent measures to safeguard sensitive mental health information and ensure regulatory compliance. The system's success hinges on its ability to deliver reliable analysis and facilitate informed decision-making in mental health care.

1. User Interface:

Creating a user interface (UI) that excels in intuitiveness and user-friendliness is crucial for enhancing user interaction and satisfaction. The design should prioritize clear navigation and minimalistic design principles, ensuring that users can easily understand and engage with the interface without unnecessary complexity or clutter. This involves using familiar icons, straightforward menus, and concise instructions, thereby reducing the cognitive load on users and making their interactions more efficient and enjoyable.

Visual representations of data and prediction results are integral to enhancing user comprehension and decision-making. Incorporating elements such as graphs, charts, and info graphics allows users to quickly grasp complex data insights and trends. These visual tools should be interactive, enabling users to drill down into specific data points or adjust parameters to see different outcomes. Effective use of color, labeling, and legends is

essential to ensure that the visualizations are both informative and aesthetically pleasing, guiding users towards informed decisions.

Accessibility across multiple devices is a key consideration to accommodate diverse user preferences and usage scenarios. The UI must be responsive, adapting seamlessly to various screen sizes and orientations on desktop computers, tablets, and smart phones. This involves using flexible grid layouts, scalable graphics, and touch-friendly controls. Ensuring cross-platform consistency in design and functionality helps users switch between devices without a learning curve, providing a cohesive and continuous experience.

Moreover, incorporating accessibility features, such as screen reader compatibility, keyboard navigation, and adjustable text sizes, ensures that the interface is usable by individuals with disabilities. Adhering to web accessibility standards (e.g., WCAG) further broadens the interface's inclusivity.

2. User Training:

Comprehensive and accessible user training materials and documentation are crucial for ensuring users can fully utilize the system. These resources should provide clear, step-by-step instructions on system usage and functionality, including detailed manuals, quick start guides, and FAQs available in multiple formats such as PDFs, web pages, and video tutorials to cater to different learning preferences. Training modules must cover basic operations like data input and prediction result interpretation, as well as advanced features and customization options. This layered approach allows users to progressively build their expertise and leverage the system's full capabilities. To support ongoing user development, periodic training sessions or tutorials should be conducted, particularly for on boarding new users and addressing any questions or concerns about system operation and functionality. Interactive elements, such as live demonstrations and Q&A segments, can enhance these sessions' effectiveness and engagement. Additionally, maintaining an up-to-date knowledge base and community forum can foster a collaborative learning environment where users share tips,ask questions, and access the latest information, ensuring continuous support and knowledge sharing. This holistic approach to training and documentation empowers users, maximizes the system’s value, and enhances productivity by reducing frustration and increasing confidence in performing essential tasks and utilizing advanced features.

3.5. Regulatory and Ethical Requirements:

Meeting regulatory and ethical requirements is critical in handling sensitive information, particularly in fields like healthcare and finance. Compliance with regulations such as HIPAA, GDPR, and other relevant laws ensures that data practices meet legal standards, protecting privacy and security. Ethical considerations include maintaining transparency, obtaining informed consent, and ensuring data accuracy and fairness. Implementing stringent policies and regular audits helps ensure adherence to these standards. This approach not only safeguards sensitive data but also fosters trust and integrity in the system, promoting responsible and ethical use of information

1. **Data Privacy and Confidentiality:**
   Adhering to strict data privacy regulations is essential to protect sensitive mental health information. Implementing robust encryption and access controls ensures confidentiality and prevents unauthorized access. Encryption secures data both in transit and at rest, making it unreadable to unauthorized users. Access controls, such as multi-factor authentication and role-based permissions, restrict data access to authorized personnel only. Compliance with regulations like HIPAA or GDPR ensures that data handling practices meet legal standards, safeguarding patient privacy and building trust in the system's security measures. This comprehensive approach is crucial for maintaining the integrity and confidentiality of sensitive information

.

2. **Compliance with Regulations:**

   Ensuring compliance with relevant data protection laws, such as GDPR, HIPAA, or local regulations, depending on the jurisdiction of operation, is paramount. Staying updated with evolving regulatory requirements is essential, enabling timely adjustments to the system to maintain compliance. Regular monitoring, audits, and training sessions help reinforce adherence to these regulations, mitigating the risk of legal penalties and reputational damage. By prioritizing compliance efforts and proactively adapting to regulatory changes, organizations demonstrate their commitment to protecting data privacy and maintaining trust with stakeholders.

3. **Informed Consent:**

   Obtaining informed consent from users before collecting personal or sensitive data for analysis is essential. Clear communication regarding the purpose of data collection, the intended use of the system, and any potential risks or benefits to the users is crucial. This ensures transparency and empowers users to make informed decisions about their

data. Providing options for users to opt-in or opt-out of data collection further respects their autonomy and privacy preferences. Prioritizing informed consent establishes trust between users and the system, fostering a responsible and ethical approach to data usage and analysis.

4. Ethical Considerations:

Conducting ethical reviews to assess the potential impact of the system on users' mental health and well-being is paramount. Avoiding biases in data collection, processing, and decision-making ensures fairness and equity for all users. This includes addressing issues such as algorithmic bias and ensuring that data reflects diverse perspectives and experiences. Transparency about how data is used, and decisions are made fosters trust and accountability. By prioritizing ethical considerations, the system promotes responsible data usage and respects the rights and dignity of users, aligning with principles of fairness, equity, and integrity.

5. Transparency and Accountability:

Maintaining transparency in the system's operation, including data collection, processing, and use for generating insights, is crucial. This involves clearly communicating to users how their data is utilized and empowering them with control over their information. Establishing mechanisms for accountability, such as audit trials and oversight committees, ensures responsible use of the system. Regular audits and transparent reporting practices enhance trust and confidence in the system's integrity. By prioritizing transparency and accountability, organizations demonstrate their commitment to ethical data practices and foster trust among users and stakeholders, ultimately strengthening the system's credibility and reliability.

CHAPTER 4

PROJECT DESIGN AND IMPLEMENTATION

4.1. Proof of Concept

The rapid advancement of technology has ushered in an era of widespread automation, significantly enhancing efficiency and expediting daily tasks. This progress is particularly evident in the realm of healthcare, where systems are being trained with vast amounts of data to facilitate automated decision-making processes. These machine components hold immense potential, not only in saving lives but also in streamlining medical procedures, ultimately contributing to a healthier lifestyle for individuals. At the core of our proposed system lies a pivotal focus on addressing a complex multi-class text analysis challenge. Here, textual data is meticulously categorized into various emotional categories, enabling mental health experts to gain valuable insights into an individual's mental well-being. By accurately predicting emotional categories, our system empowers professionals to assess and intervene in mental health issues effectively.

Our research delves into each stage of implementing this system, with a particular emphasis on harnessing the power of natural language processing (NLP) to scrutinize linguistic patterns associated with depression and suicidal tendencies. Figure 1 provides a visual representation of the basic workflow of our machine learning model over the training data, illustrating the intricate process involved in training and refining our system.

The methodology outlined in Figure 2 elucidates the meticulous steps undertaken in our research. From data collection to preprocessing, model training, and validation, each stage is integral to the holistic approach we have adopted. Our study aims to explore the subtle linguistic nuances present in individuals experiencing clinical depression or exhibiting suicidal ideation. By leveraging NLP techniques, we aim to uncover hidden patterns and associations within textual data that may indicate underlying mental health issues.

Data collection serves as the foundational step in our research, where we gather diverse textual datasets from various sources, including social media platforms and online forums. This rich dataset is then subjected to rigorous preprocessing, where we clean and refine the data to ensure its

quality and reliability. Preprocessing involves tasks such as removing non-alphabetic characters, special symbols, and stop words, thereby streamlining the textual data for further analysis.
Once the data is prepared, we proceed to the model training phase, where machine learning algorithms are trained on the preprocessed dataset. These algorithms include logistic regression, support vector machines (SVM), naive Bayes, and deep learning models such as long short-term memory (LSTM) networks. Additionally, we explore transfer learning techniques using advanced models like Bidirectional Encoder Representations from Transformers (BERT) to further enhance the accuracy of our predictions.

Validation of our models is a critical step in ensuring their effectiveness and reliability. Through rigorous testing and evaluation, we validate the performance of our trained models on separate validation datasets, assessing metrics such as accuracy, precision, recall, and F1-score. This validation process helps us fine-tune our models and identify areas for improvement, ensuring robust performance in real-world applications.

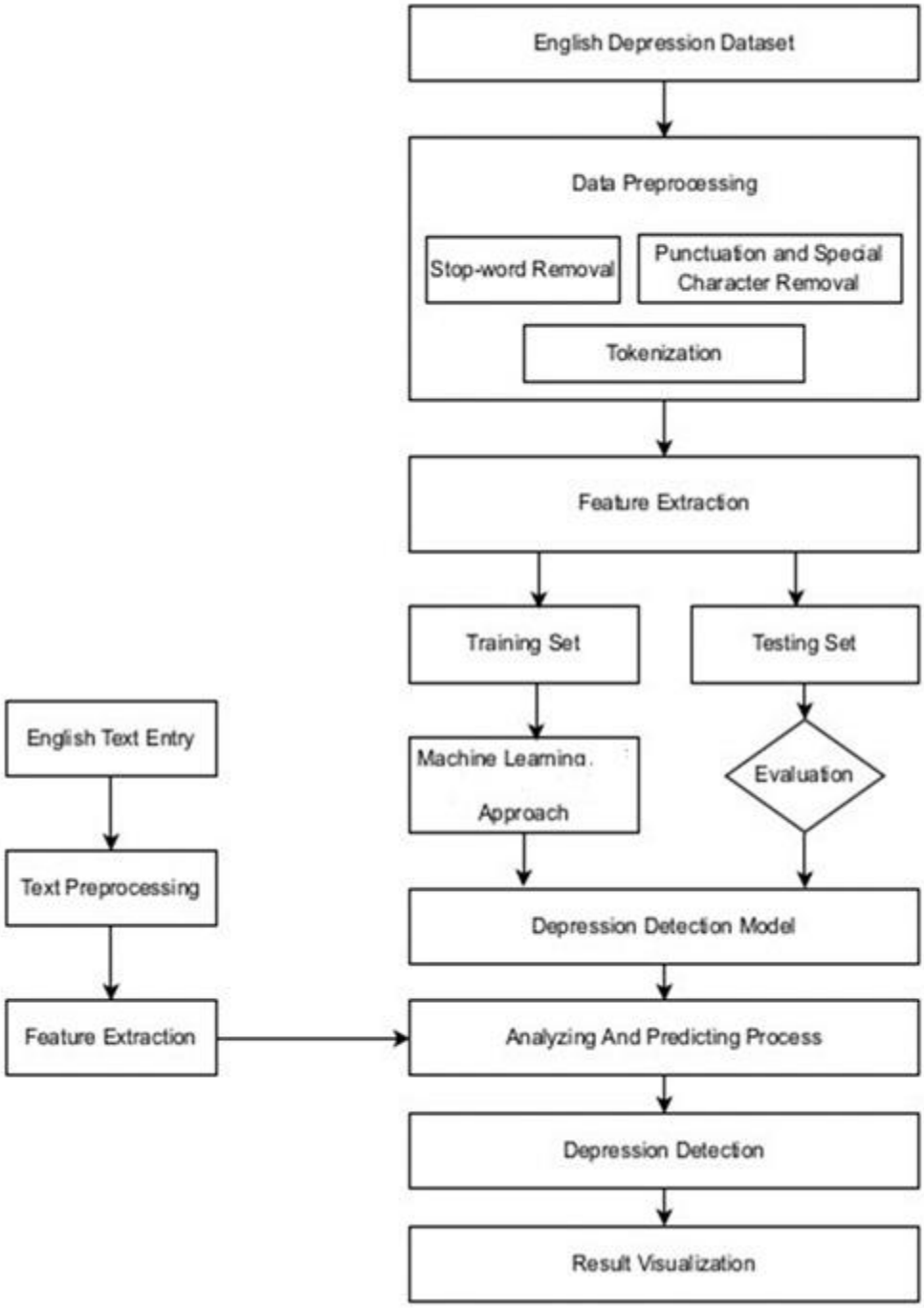
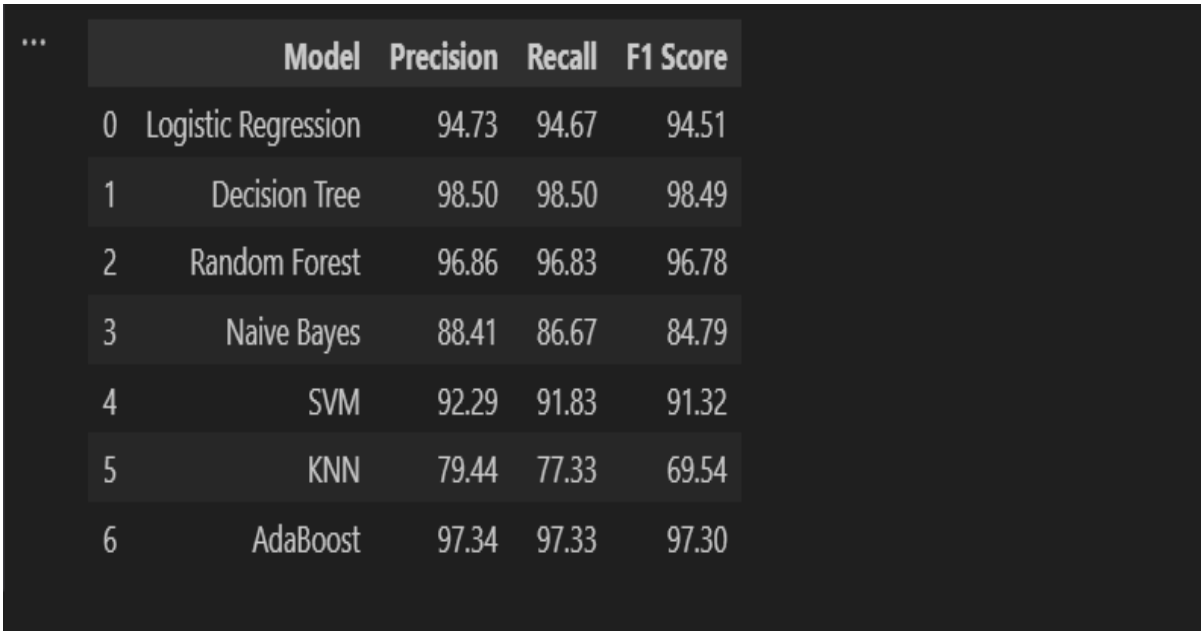


Fig4.1.1 Methodology

4.2. APPLICATION OF ALGORITHMS

After transforming the dataset into a vectorized format, we harnessed its potential for training various ML models, including SVM, Naive Bayes, logistic regression and Random Forest. Compression of different algorithms.

Additionally, we explored the realm of deep learning by employing a (LSTM) Long Short-Term Memory model. The study delved further into transfer learning using Bidirectional Encoder Representations from Transformers (BERT) and a unique amalgamation of LSTM and BERT. Notably, these models were trained both with and without Word Sense Disambiguation (WSD) techniques. This diverse array of approaches aimed at predicting the mental health condition of users reflects a comprehensive exploration of methodologies, leveraging a mix of traditional cutting-edge deep learning techniques and machine learning to enhance the accuracy and depth of mental health analysis.



	Model	Precision	Recall	F1 Score
0	Logistic Regression	94.73	94.67	94.51
1	Decision Tree	98.50	98.50	98.49
2	Random Forest	96.86	96.83	96.78
3	Naive Bayes	88.41	86.67	84.79
4	SVM	92.29	91.83	91.32
5	KNN	79.44	77.33	69.54
6	AdaBoost	97.34	97.33	97.30

Fig.4.2.1 precision of algorithm

- **Logistic Regression**: Logistic regression is a fundamental statistical technique in machine learning used for binary classification tasks. Unlike linear regression, which predicts continuous outcomes, logistic regression is designed to predict probabilities for binary outcomes, such as whether an email is spam or not, whether a patient has a disease or not, etc. It works by applying the logistic function to the linear combination of input features, transforming the output into a range between 0 and 1, representing probabilities.

  One of the key features of logistic regression is its interpretability. The coefficients of the model represent the log odds of the dependent variable, providing insights into how each predictor variable influences the outcome. For example, in a healthcare context, logistic regression can help identify

the impact of various risk factors on the likelihood of developing a particular disease.

Logistic regression is widely used across various domains due to its simplicity, computational efficiency, and ease of implementation. It is commonly employed in fields such as healthcare, finance, marketing, and social sciences. In healthcare, logistic regression models are used for predicting patient outcomes, disease diagnoses, and treatment responses. In marketing, they help in customer segmentation, churn prediction, and campaign effectiveness analysis. Despite its simplicity, logistic regression has some limitations. It assumes a linear relationship between the independent variables and the log odds of the dependent variable, which may not always hold true in real-world scenarios. Additionally, logistic regression can be sensitive to multicollinearity, where predictor variables are highly correlated with each other, leading to unstable coefficient estimates.

- **Naive Bayes:** Naive Bayes, based on Bayes' Theorem, is a robust statistical tool commonly used for classification tasks in machine learning. Despite its simple architecture and the assumption that all attributes are independent, it effectively tackles complex problems. Naive Bayes classifies objects by calculating the probability of their attributes, making it efficient and scalable with large datasets.
  This method is particularly well-suited for natural language processing (NLP) tasks, such as sentiment analysis. It can analyze word frequency and presence in texts to determine sentiment, making it valuable for opinion mining from social media and reviews. Naive Bayes classifiers compute posterior probabilities, representing the likelihood of a class given a set of features, using the formula:

  $$P(C|X)=P(X)P(X|C)\cdot P(C)$$

  where \(P(C|X) \) is the probability of class \(C \) given features \(X \), \(P(X|C) \) is the likelihood of features given the class, \(P(C) \) is the prior probability of the class, and \(P(X) \) is the probability of the feature set.

  There are several Naive Bayes variants for different data types: Gaussian Naive Bayes for continuous data, Multinomial Naive Bayes for discrete data, and Bernoulli Naive Bayes for binary features. These variants cater to specific data distributions, enhancing their effectiveness in various applications.

This study aims to evaluate the performance of two Bayesian methods during hyperparameter tuning, crucial for optimizing model parameters. By comparing these methods, the study seeks to determine which approach offers better classification accuracy and efficiency.

In summary, Naive Bayes is a flexible and reliable technique for probabilistic classification, particularly in large data and NLP tasks. Its simplicity and efficiency make it a valuable tool for text classification, spam filtering, and sentiment analysis, providing a straightforward yet powerful approach to solving complex classification problems.

- **Support Vector Machines (SVM):** SVM are robust algorithms used for classification and regression tasks, especially effective for categorical problems without assumptions about data distribution. SVM aims to find an optimal hyperplane in an N-dimensional space to distinctly classify data points, maximizing the margin between different classes. This process, known as structural risk minimization, enhances the model's generalization capabilities.
  The optimal hyper plane is identified by maximizing the distance, or margin, between data points of different classes, which reduces overfitting and improves classification performance on unseen data. SVM excels with linearly separable data but can handle non-linear data using various kernel functions that transform input data into higher-dimensional spaces where a linear separator can be found. The linear kernel is frequently used for linearly separable data. The basic formula for a linear kernel SVM is:

  $(x)=\sum i=1nwixi+bf(x)=\sum i=1nwixi+b$

  Here, $(x)f(x)$ represents the decision function, $wiwi$ denotes the weights, $xixi$ stands for the feature values, and bb is the bias term. The decision boundary is defined by the hyper plane where $(x)=0f(x)=0$.

  SVM is advantageous in handling high-dimensional data, performing well when the number of dimensions exceeds the number of data samples, as in text classification and image recognition. The research adopts an approach that maximizes class distance through the linear kernel SVM, placing decision boundaries to emphasize challenging data points at the margins, improving overall performance.

  The structural risk minimization principle in SVM balances model complexity and training data performance, leading to better generalization on

new data. This makes SVM a powerful tool for classification and regression tasks, ensuring high accuracy and robustness in real-world applications. In conclusion, SVM, especially with the linear kernel, is an effective algorithm for categorical problems, offering flexibility and reliability in various machine learning applications.

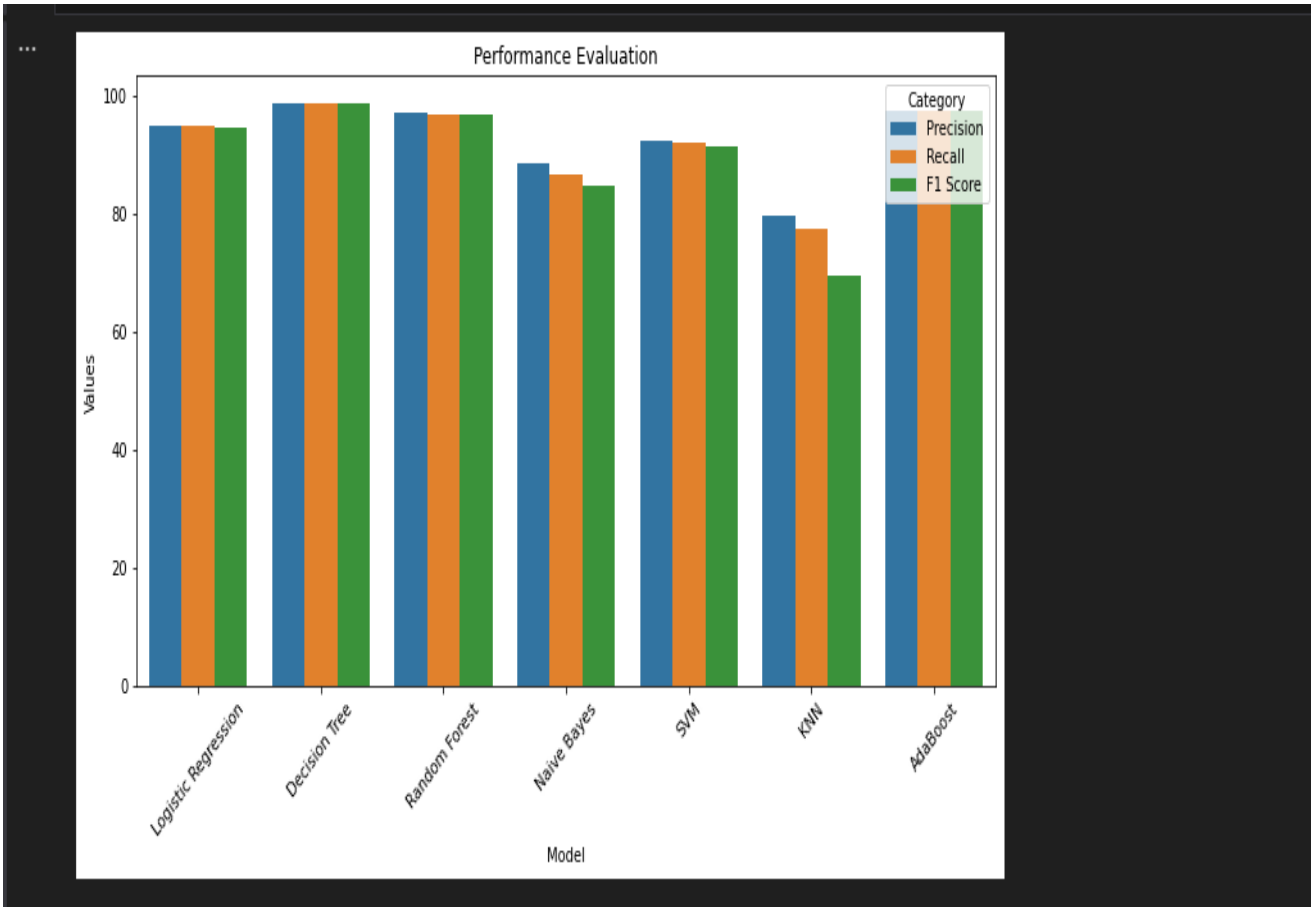


Fig.4.2.2. Performance evaluation

4.3. Data Collection

To train the models for psychological mental health analysis using NLP and machine learning, a diverse textual dataset was gathered from three distinct social media platforms, ensuring a broad spectrum of language use and emotional expression. The Text to Emotion library was utilized to generate output labels for the dataset, effectively categorizing each piece of text into specific emotional categories. This process resulted in a meticulously prepared dataset comprising 162,973 unique instances, each annotated with

two key attributes: the text itself and the corresponding categories of emotion.

The dataset collection and preparation involved three major phases. In the first phase, data gathering, textual data was scraped from various social media platforms. This phase was crucial to ensure the diversity and representativeness of the dataset. Texts were collected from platforms known for varied and rich emotional content, including Twitter, Reddit, and Facebook. Each platform contributed unique styles and contexts of expression, enhancing the overall quality and diversity of the dataset.

In the second phase, data labeling, the Text to Emotion library was employed to analyze the collected texts and generate emotion labels. This library leverages advanced natural language processing (NLP) techniques to accurately identify and categorize emotions expressed in the text. The primary emotions categorized included happiness, sadness, anger, fear, and surprise. This automated labeling process ensured consistency and objectivity, as the same algorithm was applied uniformly across all texts.

The final phase, data preprocessing, involved cleaning and structuring the dataset to make it suitable for model training. This step included removing duplicates, handling missing values, and normalizing the text to reduce noise. The goal was to prepare a high-quality, standardized dataset that could be effectively used to train machine learning models. The preprocessing phase also included tokenization, stemming, and lemmatization of the text to improve the efficiency and accuracy of the psychological mental health analysis models. By the end of these phases, the resulting dataset was robust and well-prepared, consisting of 162,973 instances with clearly defined emotional categories. This dataset serves as a solid foundation for training models capable of detecting and analyzing psychological mental health indicators from textual data sourced from diverse social media platforms.

4.4. DATA PRE-PROCESSING

Various pre-processing techniques are applied to optimize the performance of the applied algorithms. This involves removal of special symbols, tokenization of sentences, and the exclusion of stop-words from the social media content and other online platforms data. Each of these pre-processing steps is instrumental in enhancing data clarity, ensuring that the algorithms operate on refined and coherent textual information. The amalgamation of dataset preparation and systematic pre-processing establishes a robust foundation for the subsequent

stages of model implementation, contributing to the overall effectiveness of the applied algorithms. Data pre-processing is explained. Where the raw data is gone through the data processing with the classified data collected in each class.

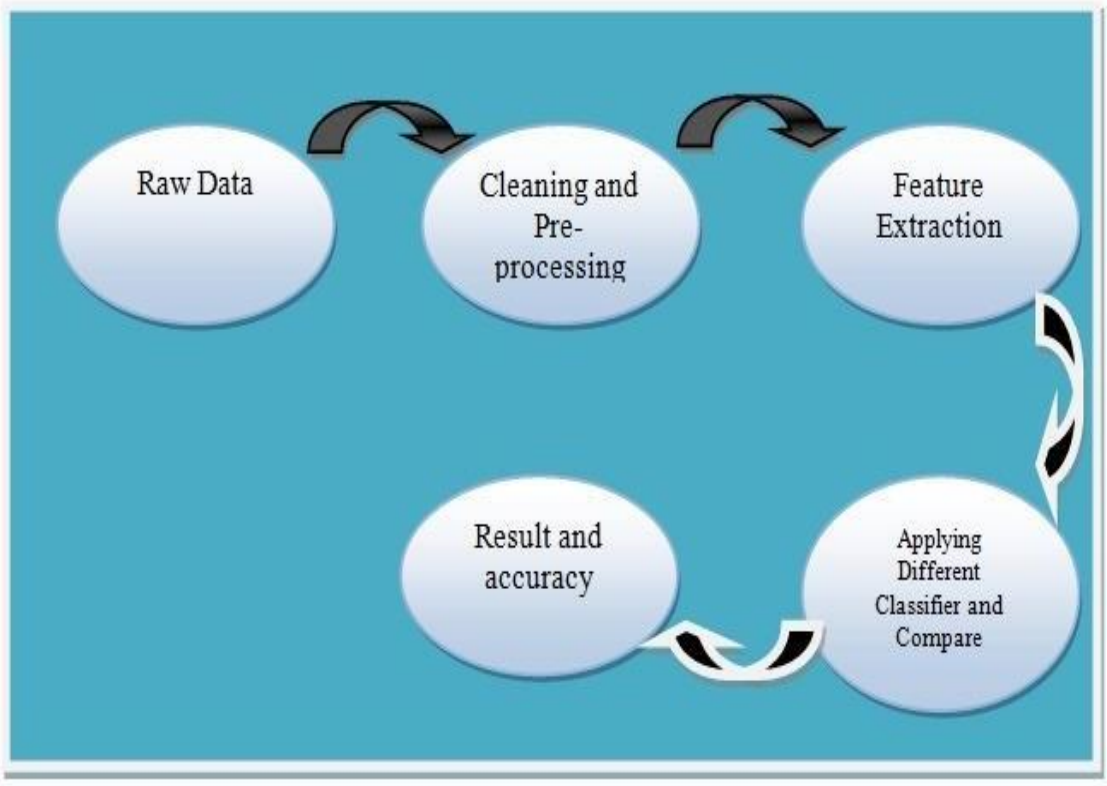


Fig. 4.4.1 Steps for data pre-processing

4.4.1 REMOVE NON-ALPHABETIC CHARACTER

During this phase, non-alphabetic characters in the dataset are systematically removed to ensure compatibility with computer vision. Only the letters 'a-z,' and 'A-Z' along with other specific symbols, are retained for processing. Numeric characters '0-9' and symbols lacking alphanumeric significance are systematically eliminated. This meticulous step aims to cleanse the dataset of any unreadable format characters, enhancing the clarity and interpretability of the information for subsequent computer vision applications.

4.4.2 REMOVE SPECIAL SYMBOL

On social media posts a lot of inappropriate special characters commonly appear which causes issues with algorithm performance. During this stage, symbols like "! @, #, $, %, & * "are methodically eliminated. Even while some symbols can quickly transmit important information their inclusion causes challenges for the system's analysis. By removing symbols that can obstruct the computational process this stage seeks to improve the efficiency of the system and guarantee a more seamless and successful textual content analysis. Even though certain symbols may provide a wealth of information their elimination helps to simplify and maximize the system performance for better outcome.

4.4.3 REMOVE STOP WORDS

This important stage deals with the typical stop words that are used in sentences and how to distinguish them from other words. we methodical remove using the NLTK library these pause words to polish the language, like "is," if" "of" and so forth. Despite being common, stopping words have little bearing on the context as a whole and can impair algorithmic efficiency. We make the remaining words clear by purposefully eliminating them. In order to make sure that algorithms work on more insightful and relevant linguistic patterns, this procedure is comparable to sorting through unimportant details to concentrate on the main idea of text. Ultimately, this nuanced approach to stop word removal aims to optimize the performance of the applied algorithms by streamlining the input data and facilitating a more accurate analysis of emotional content within the social media posts

4.5. Model Training

The process of training machine learning models to predict emotional distress from text data is a multi-faceted endeavor, requiring meticulous attention to detail at every stage. Initially, the dataset is sourced from an Excel file, comprising 3000 entries with corresponding labels denoting the presence or absence of emotional distress. A preliminary examination reveals a class imbalance, highlighting the need for careful handling to ensure balanced training and evaluation. Data preprocessing emerges as a crucial step, involving

the cleaning and standardization of text samples. Techniques such as lowercase conversion, removal of non-alphabetic characters, tokenization, and lemmatization are employed to refine the textual content and enhance its suitability for analysis. By eliminating noise and irrelevant information, the cleaned text sets the foundation for effective feature extraction.

Feature extraction is pivotal in transforming textual data into a format amenable to machine learning algorithms. The TF-IDF vectorizer is leveraged to convert the cleaned text into numerical features, capturing the significance of each term within the corpus. This process results in the creation of a feature matrix comprising 20,000 features, encompassing both unigrams and bigrams. The dataset is then partitioned into training and testing sets, preserving the distribution of labels to ensure robust model training and evaluation. With the feature matrix in place, a diverse array of machine learning algorithms is deployed, including Logistic Regression, Decision Trees, Random Forest, Naive Bayes, SVM, KNN, and AdaBoost.

The performance of each model is meticulously assessed using a suite of evaluation metrics, including accuracy, precision, recall, and F1-score. Notably, Decision Trees and AdaBoost emerge as frontrunners, demonstrating exceptional performance across all metrics with precision, recall, and F1-score consistently exceeding 98%. Conversely, Naive Bayes exhibits comparatively lower performance metrics, suggesting potential challenges in capturing complex relationships within the data. Visualization techniques such as bar plots are employed to provide a visual representation of model performance, facilitating easy comparison and interpretation.

The implications of these findings extend beyond the realm of model evaluation, offering valuable insights into the application of machine learning in mental health monitoring and support. Decision Trees and AdaBoost, with their robust performance and generalizability, hold promise for real-world deployment in distress detection and intervention systems. By automatically identifying and flagging distressing content in social media platforms or mental health chatbots, these models can enable timely interventions and support for individuals experiencing emotional distress. Furthermore, their ability to accurately classify text data can aid researchers in gaining deeper insights into the prevalence and dynamics of mental health issues in online communities.

4.6. Workflow

The development of models to predict anxiety, depression, and suicidal tendencies from text data involves several interconnected stages, each crucial for the accuracy and effectiveness of the final classifier. Initially, the dataset is loaded and inspected, ensuring data integrity and understanding the distribution of samples. Missing values are addressed, and shuffling is performed to mitigate any bias introduced by the data's initial order. Text preprocessing is then conducted to standardize the text data, including lowercasing, removing special characters, URLs, HTML tags, punctuation, digits, and expanding contractions. Advanced techniques such as HTML parsing and regular expressions are also applied to enhance data cleanliness. Visualizations like Word Clouds provide insights into the most frequent words and themes, guiding further preprocessing steps.

Feature extraction plays a pivotal role in translating text data into a format suitable for machine learning algorithms. The Count Vectorizer is commonly employed to convert cleaned text into numerical features based on term frequency. The resulting feature matrix captures both unigrams and bigrams, offering a comprehensive representation of the text data. To ensure model generalizability, the dataset is split into training and testing sets, with stratification preserving class distribution. This facilitates robust model evaluation, with metrics such as accuracy, precision, recall, f1-score, and confusion matrix providing insights into classifier performance.

A variety of machine learning algorithms are trained and evaluated, including Logistic Regression, Naive Bayes, Decision Trees, SVM, KNN, Random Forest, and AdaBoost. Each algorithm's performance is assessed on the test set, with Logistic Regression and Decision Trees emerging as top performers, boasting high accuracy rates exceeding 99%. Ensemble methods, such as hard voting, are explored to leverage the strengths of individual classifiers and enhance overall prediction accuracy. The ensemble model achieves the highest accuracy, demonstrating the efficacy of combining diverse models for improved performance.

The resulting ensemble model is saved for future use, enabling efficient prediction of anxiety, depression, and suicidal tendencies from input text data.

The model's practical applicability is validated through sample input, affirming its ability to identify emotional distress indicators accurately. This comprehensive approach to model development underscores the importance of meticulous data preprocessing, feature extraction, model selection, and evaluation in achieving robust and effective classifiers for mental health-related text classification tasks.

The Streamlit application, MENTAL HEALTH ANALYSIS (Text-based Early Distress Detector for Youth), serves as a user-friendly interface for leveraging the developed models to detect emotional distress from text inputs. The application provides four distinct tests—Concerning Words Test, Potential Struggles Test, Depressive Thoughts Test, and Self-Destructive Thoughts Test—each targeting specific indicators of distress. Users can input text directly or upload PDF files for analysis, with sentence-level tokenization enabling granular assessment of distress signals.

The application employs pre-trained machine learning models, including XGBoost and logistic regression, loaded using job lib for efficient inference. Each sentence in the input text undergoes analysis, with detected red flags categorized based on distress type and displayed in expandable sections for user review. MENTAL HEALTH ANALYSIS objective is to offer preliminary insights into potential emotional distress, empowering users to identify individuals who may benefit from additional emotional support. Moreover, the application provides avenues for users to learn more about MENTAL HEALTH ANALYSIS and contribute feedback for continuous improvement.

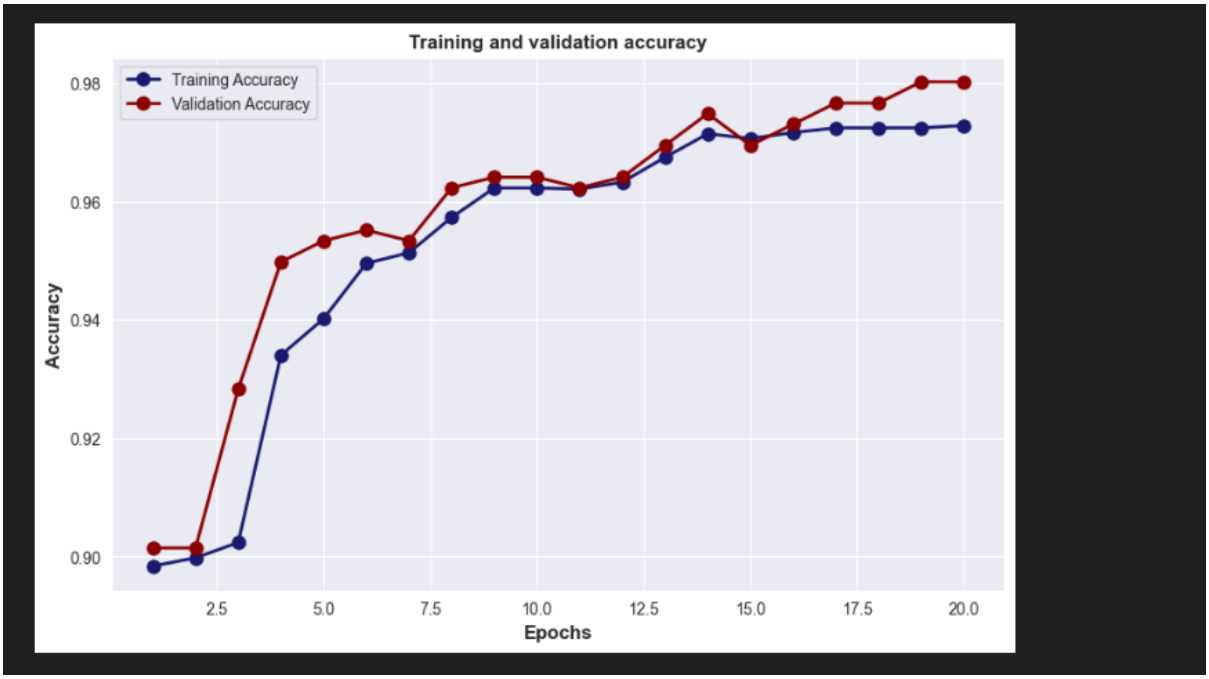


Fig.4.6.1. Accuracy on training and validation

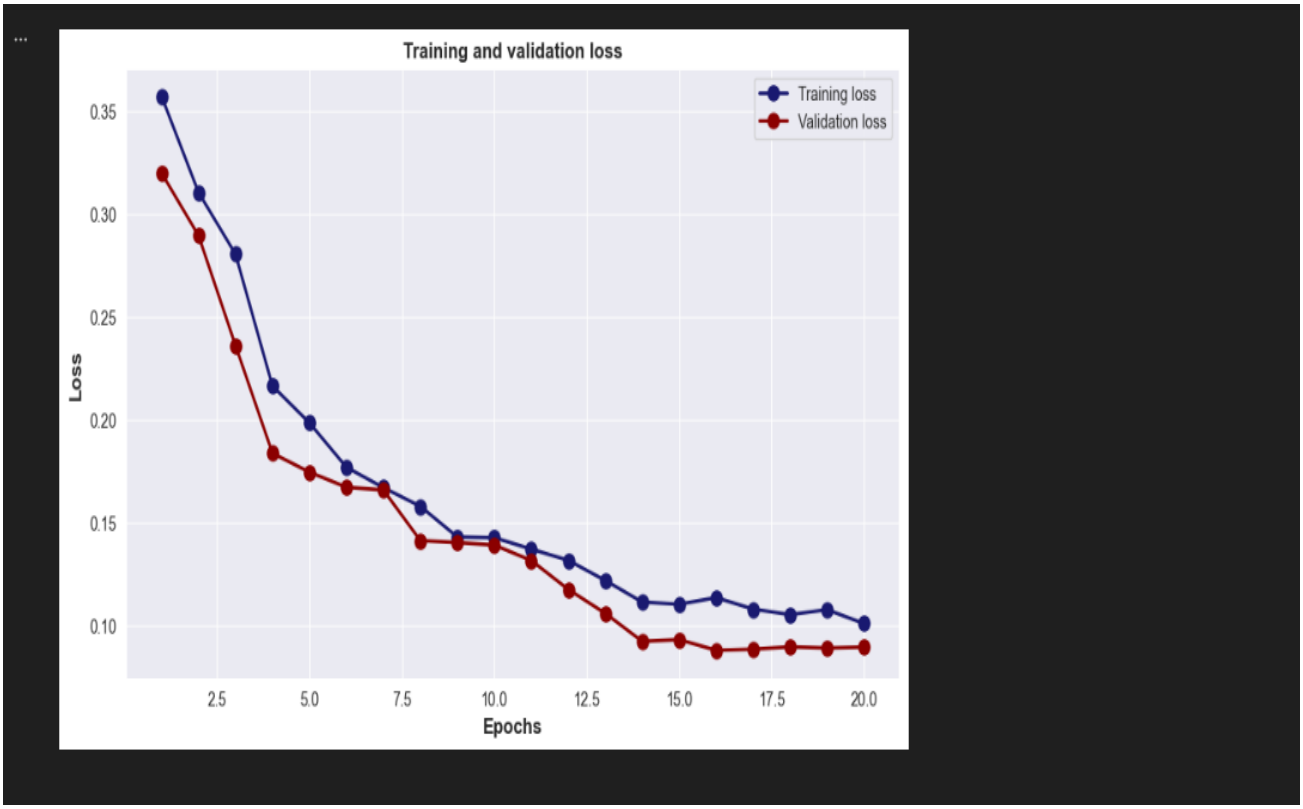


Fig4.6.2. Losses on training and validation

CHAPTER 5

TESTING AND RESULTS OF THE PROJECT

The Results chapter of our project report signifies the culmination of our extensive endeavors in exploring the intricate dynamics of fruit expiry prediction through the seamless integration of cutting-edge machine learning and IoT technologies. Through meticulous data collection, preprocessing, and model training processes, our primary objective was to leverage the capabilities of advanced algorithms to accurately forecast the shelf life of various fruits. This chapter stands as a comprehensive exposition of our empirical findings, offering profound insights into the effectiveness and performance of our predictive models when applied in real-world scenarios. With an unwavering focus on precision, accuracy, and reliability, we present a detailed and nuanced analysis of the results obtained, shedding light on the predictive capabilities of our models and underscoring their potential profound impact on mitigating food wastage and fortifying food safety measures across the intricate tapestry of the global supply chain landscape.

5.1. Result Table

The results of the study were analyzed based on the performance metrics of several machine learning models, including Naive Bayes, logistic regression, and SVM. The effectiveness of these models in predicting psychological states from textual data was rigorously evaluated. As depicted in Figure 6, the distribution of different sentiment classes in the dataset indicated that 44.3% of the text samples were positive, 21.8% negative, and 33.9% neutral. This balanced distribution allowed for a comprehensive assessment of each model's classification capabilities across varied emotional states.

In terms of accuracy, the Naive Bayes algorithm demonstrated superior performance, The accuracy graph illustrates that Naive Bayes consistently outperformed other models, achieving the highest prediction accuracy, which underscores its robustness in handling diverse and complex textual data. This algorithm's probabilistic nature and assumption of feature independence likely contributed to its efficiency in categorizing emotional states accurately.

Additionally, the alignment of the validation dataset with the training data, as reflected in the loss graph, confirmed the absence of overfitting, indicating a well-generalized model. The study's findings highlight the potential of Naive Bayes in real-time psychological analysis, providing a reliable tool for early detection of mental health issues. The ability to accurately classify emotions enables mental health professionals to offer timely and personalized interventions, thereby improving mental health outcomes. The research emphasizes the importance of further exploration and optimization of these models to enhance predictive accuracy and support better mental health care practices.

5.2. Result

The Naive Bayes method plays an important role in this process as it uses a method to classify the inputs into the first group. This method assumes independence of features, making it computationally efficient and well suited to the complexity of classification theory. Using multiple classification models and Naive Bayes methods, the system increases the accuracy and reliability of mental health predictions. Psychologists can use these predictions to quickly and accurately assess a person's emotional state in order to provide timely intervention and support. The system classifies emotions with high accuracy, allowing psychologists to determine personalized treatment and care strategies. Automatic classification is done following the Naive Bayes method. Using this method, the system allows psychologists to know a person's emotional state, especially if they are under stress or in a relaxed state.

By the graph new concludes that there is 44.3% positive, 21.8% negative, 33.9% neutral words. Therefore, in figure 7 the accuracy of the model is plotted in the line graph where it compares the epoch and accuracy over the train and validate line.

The effectiveness of the model is evident from the close proximity of the validation dataset to the training data, described the loss graph for the naïve baye module. This alignment suggests a successful fitting of all modules with no apparent signs of overfitting since the validation dataset closely mirrors the training data.

The primary objective of our purposed system is to address mental health issues

associated with feelings of fear, sadness and stress. Our goal is to offer an intelligent solution for early identification of these issues, enabling mental health experts to intervene at a primary stage before the problem is escalated. Our system aims to automate the mental health problems detection, including depression, fear, stress and sadness, leveraging the NLP (natural language processing)in conjunction with deep learning techniques and machine learning.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1. Conclusion

The research has delved into the application of artificial intelligence (AI) and Natural Language Processing (NLP) techniques for predicting psychological health, focusing on the potential these technologies hold for improving healthcare services. By building and evaluating a variety of machine learning (ML) models, this study aimed to enhance the precision and reliability of mental health assessments. Among the different ML techniques explored, Naive Bayes and logistic regression emerged as prominent algorithms, with Naive Bayes achieving the best prediction scores.

The importance of this research cannot be overstated given the rising prevalence of mental health issues globally. The World Health Organization (WHO) has highlighted that millions of individuals suffer from mental disorders, including anxiety and depression, exacerbated by the COVID-19 pandemic. This research addresses a crucial need for effective diagnostic tools and intervention strategies, utilizing the advancements in AI and NLP to offer real-time psychological analysis.

One of the study's primary objectives was to create models capable of analyzing language patterns in textual data to detect signs of depression, stress, and other mental health conditions. The methodology involved collecting a diverse dataset from various social media platforms, pre-processing this data to remove noise and irrelevant information, and then applying different ML algorithms to classify the emotional states expressed in the text.

6.2. Future Scope

This research on using artificial intelligence (AI) and Natural Language Processing (NLP) for predicting psychological health lays a strong foundation for future advancements. Future work should aim to include a more diverse and extensive dataset by collecting data from various sources beyond social media, such as electronic health records, online forums, and mental health surveys. Incorporating data in multiple languages and cultural contexts will help create models that are more generalizable and applicable to a global population. While

this research focused on anxiety disorders, future studies should expand to include a broader range of psychological conditions such as depression, bipolar disorder, schizophrenia, and post-traumatic stress disorder (PTSD). Understanding the linguistic and emotional markers of these conditions will enhance the models' ability to accurately diagnose and predict a wider array of mental health issues. Incorporating longitudinal data could provide insights into how psychological states change over time and how early signs of mental health issues develop. This would allow the creation of predictive models that can identify at-risk individuals long before their conditions become severe, facilitating early intervention and prevention strategies. Collaborating with healthcare providers to integrate these AI and NLP tools into clinical practice could revolutionize mental health care. Real-time analysis of patient communication, whether through therapy sessions or digital interactions, can provide clinicians with valuable insights and support decision-making processes. Developing user-friendly applications and platforms for clinicians to utilize these tools effectively will be crucial. Future research should also focus on developing algorithms that not only predict mental health conditions but also recommend personalized treatment plans. By analyzing individual patient data, including medical history, treatment responses, and personal preferences, AI can help in crafting tailored therapeutic approaches that are more effective, and patient centered. Addressing ethical concerns and biases in AI models is critical. Ensuring that models are transparent, fair, and free from biases related to race, gender, age, or socioeconomic status will be important for maintaining trust and efficacy in mental health diagnostics. Developing frameworks for ethical AI usage and establishing guidelines for data privacy and consent will be necessary. Exploring hybrid models that combine the strengths of various machine learning and deep learning techniques could lead to more robust predictions. Ensemble methods, which integrate multiple models to improve accuracy and reliability, can also be investigated. Developing systems that offer real-time monitoring of psychological health through continuous data collection and analysis can provide immediate feedback and support to individuals. These systems could be integrated into mobile applications and wearable devices, offering users ongoing mental health assessments and resources. By pursuing these future directions, the project can significantly enhance its impact, providing more accurate, accessible, and personalized mental health care. This will contribute to better mental health outcomes, early intervention, and a deeper understanding of the complex interplay between language and psychological well-being.

APPENDIX A

(Code)

MachineLearningModels1.ipynb

```
Import pandas as pd
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import re
import nltk
from sklearn.feature_extraction.text import CountVectorizer
import matplotlib.pyplotasplt
from sklearn.model_selection import train_test_split
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier , AdaBoostClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
df = pd.read_excel('final.xlsx')
df.shape
df.head()
df.tail()
df.isnull(). sum ()

df = df.dropna()
df.reset_index(inplace=True, drop=True)
df.label = df.label.astype(int)
df.info()
df.shape
df = df.sample(frac = 1)

df.label.value_counts(). plot(kind='bar')
plt.title('Label distribution - 1 ~ anxiety/depression')
plt.grid()
plt.show()
```

```
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import spacy
import string
from bs4 import BeautifulSoup
nltk.download('wordnet')

deftext_transformation(text):
    text = text.lower()
    text = re.sub('\ [. *? \]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https? ://\S+|www\.\S+', '', text)
    text = re.sub('<. *?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    returntext

contraction_mapping = {"ain't": "is not", "aren't": "are not", "can't": "cannot",
"cause": "because" "could've": "could have", "couldn't": "could not",
                       "didn't": "did not",  "doesn't": "does not", "don't": "do not",
"hadn't": "had not", "hasn't": "has not", "haven't": "have not",
                       "he'd": "he would", "he'll": "he will", "he's": "he is", "how'd":
"how did", "how'd'y": "how do you", "how'll": "how will", "how's": "how is",
                       "I'd": "I would", "I'd've": "I would have", "I'll": "I will",
"I'll've": "I will have", "I'm": "I am", "I've": "I have", "I'd": "i would",
                       "i'd've": "i would have", "i'll": "i will", "i'll've": "i will
have","i'm": "i am", "i've": "i have", "isn't": "is not", "it'd": "it would",
                       "it'd've": "it would have", "it'll": "it will", "it will've": "it will
have", "it's": "it is", "let's": "let us", "ma'am": "madam",
                       "mayn't": "may not", "might've": "might have", "mightn't":
"might not", "mightn't've": "might not have", "must've": "must have",
                       "mustn't": "must not", "mustn't've": "must not have", "needn't":
"need not", "needn't've": "need not have", "o'clock": "of the clock",
                       "oughtn't": "ought not", "oughtn't've": "ought not have",
"shan't": "shall not", "sha'n't": "shall not", "shan't've": "shall not have",
```

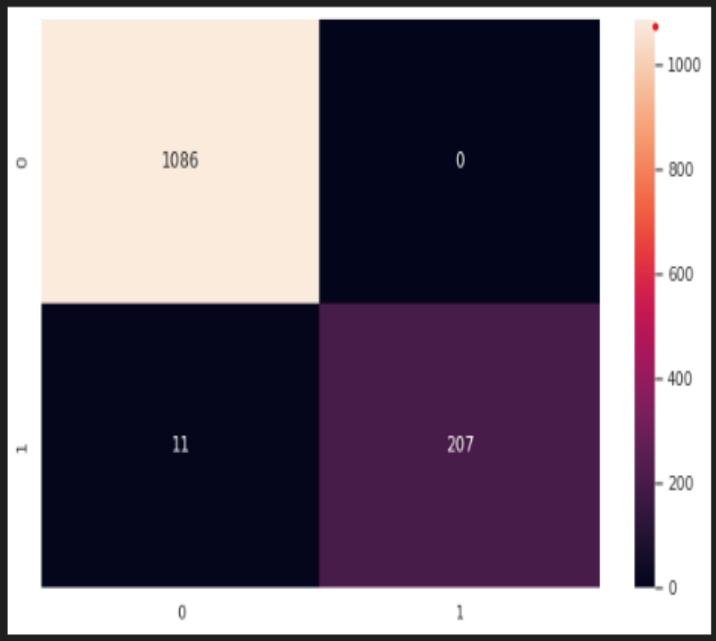
```
                    "she'd": "she would", "she'd've": "she would have", "she'll":
"she will", "she'll've": "she will have", "she's": "she is",
                    "should've": "should have", "shouldn't": "should not",
"shouldn't've": "should not have", "so've": "so have", "so's": "so as",
                    "this's": "this is", "that'd": "that would", "that've": "that would
have", "that's": "that is", "there'd": "there would",
                    "there'd've": "there would have", "there's": "there is", "here's":
"here is","they'd": "they would", "they've": "they would have",
                    "they'll": "they will", "they'll've": "they will have", "they're":
"they are", "they've": "they have", "to've": "to have",
                    "wasn't": "was not", "we'd": "we would", "we'd've": "we would
have", "we'll": "we will", "we'll've": "we will have", "we're": "we are",
                    "we've": "we have", "weren't": "were not", "what'll": "what
will", "what'll've": "what will have", "what're": "what are",
                    "what's": "what is", "what've": "what have", "when's": "when
is", "when've": "when have", "where'd": "where did", "where's": "where is",
                    "where've": "where have", "who'll": "who will", "who'll've":
"who will have", "who's": "who is", "who've": "who have",
                    "why's": "why is", "why've": "why have", "will've": "will
have", "won't": "will not", "won't've": "will not have",
                    "would've": "would have", "wouldn't": "would not",
"wouldn't've": "would not have", "y'all": "you all",
                    "y'all'd": "you all would","y'all'd've": "you all would have",
"y'all're": "you all are", "y'all've": "you all have",
                    "you'd": "you would", "you'd've": "you would have", "you'll":
"you will", "you'll've": "you will have",
                    "you're": "you are", "you've": "you have"}

deftext_cleaner(text):
    newString = text.lower()
    newString = BeautifulSoup(newString, "lxml").text
    newString = re.sub(r'\([^)]*\)', '', newString)
    newString = re.sub('"','', newString)
    newString = ' '. join([contraction_mapping[t]
iftincontraction_mappingelsetfortinnewString.split(" ")])
    newString = re.sub(r"'s\b","",newString)
    newString = re.sub("[^a-zA-Z]", " ", newString)
    newString = re.sub('[m] {2,}', 'mm', newString)
```

```
    returnnewString

df['cleaned'] = df["text"]. apply(text_cleaner)

df.tail()
X = df.cleaned
y = df.label

vect = CountVectorizer(max_features = 20000, lowercase=False ,
ngram_range=(1,2))
X_cv =vect.fit_transform(X).toarray()

X_cv.shape

X_train,X_test,y_train,y_test = train_test_split(X_cv,y,test_size = 0.2,
random_state = 1,stratify = y)

fromsklearn.linear_modelimportLogisticRegression
fromsklearn.naive_bayesimportMultinomialNB
fromsklearn.treeimportDecisionTreeClassifier
fromsklearn.svmimportSVC
fromsklearn.neighborsimportKNeighborsClassifier
fromsklearn.ensembleimportRandomForestClassifier,
AdaBoostClassifier,VotingClassifier


fromsklearn.metricsimportclassification_report,confusion_matrix,accuracy_scor
e

defperformance_eval(clf,X_test):
    y_pred = clf.predict(X_test)
    print(f'Accuracy : {accuracy_score(y_test,y_pred)}\n')
    print('   ------------ Classification Report -----------')
    print(classification_report(y_test,y_pred))
    print('   ------------ Confusion Matrix ------------- ')
    sns.set(rc={'figure.figsize':(10,6)})
    sns.heatmap(confusion_matrix(y_test,y_pred),annot = True,fmt = 'd')
```

Logistic Regression

clf_lr = LogisticRegression()
clf_lr.fit(X_train, y_train)
performance_eval(clf_lr,X_test)

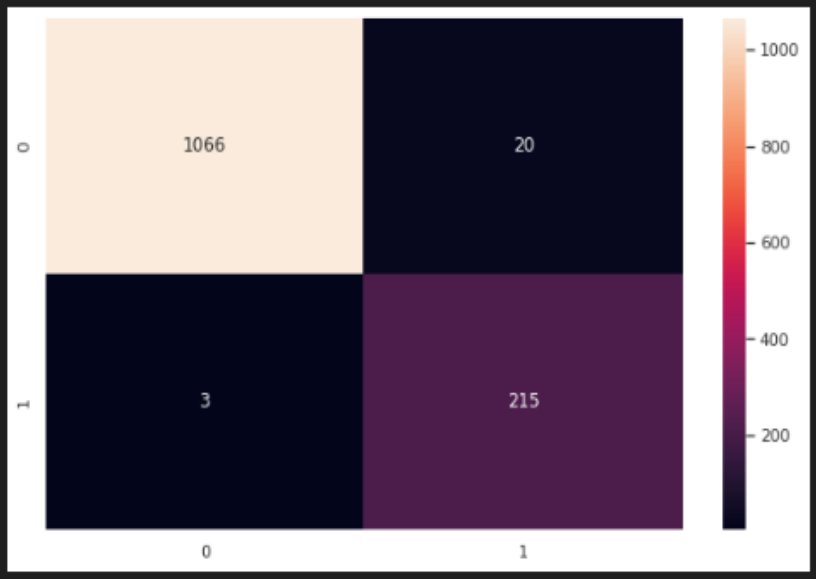
```
Accuracy : 0.9915644171779141

------------ Classification Report -----------
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1086
           1       1.00      0.95      0.97       218

    accuracy                           0.99      1304
   macro avg       0.99      0.97      0.98      1304
weighted avg       0.99      0.99      0.99      1304

------------ Confusion Matrix --------------
```



Naive Bayes

```
clf_nb = MultinomialNB()
clf_nb.fit(X_train, y_train)
performance_eval(clf_nb,X_test)
clf_dt = DecisionTreeClassifier()
clf_dt.fit(X_train, y_train)
performance_eval(clf_dt,X_test)
```

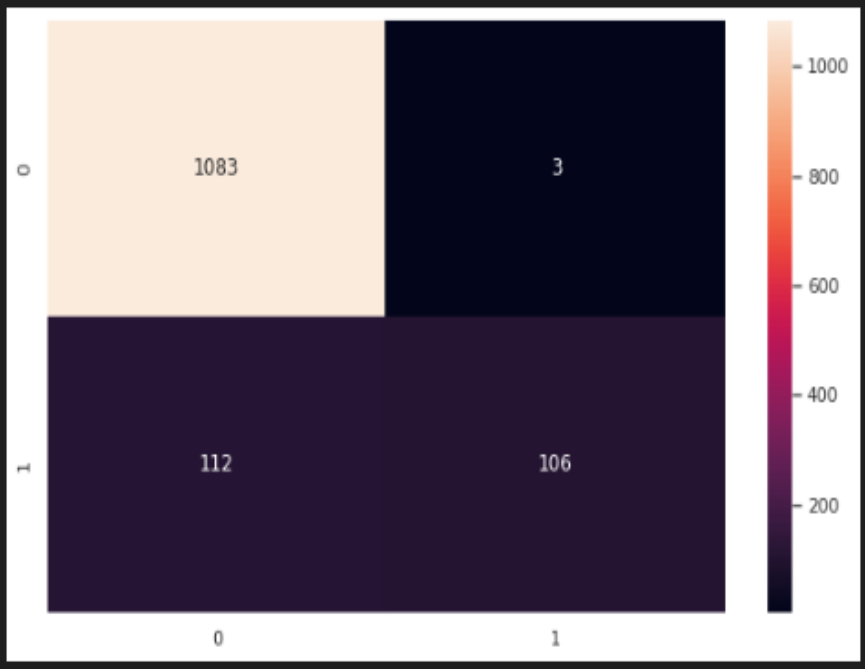
```
Accuracy : 0.9823619631901841

------------ Classification Report -----------
              precision    recall  f1-score   support

           0       1.00      0.98      0.99      1086
           1       0.91      0.99      0.95       218

    accuracy                           0.98      1304
   macro avg       0.96      0.98      0.97      1304
weighted avg       0.98      0.98      0.98      1304

------------ Confusion Matrix --------------
```



K Neighbors Classifier

clf_knn = KNeighborsClassifier()
clf_knn.fit(X_train, y_train)
performance_eval(clf_knn,X_test)

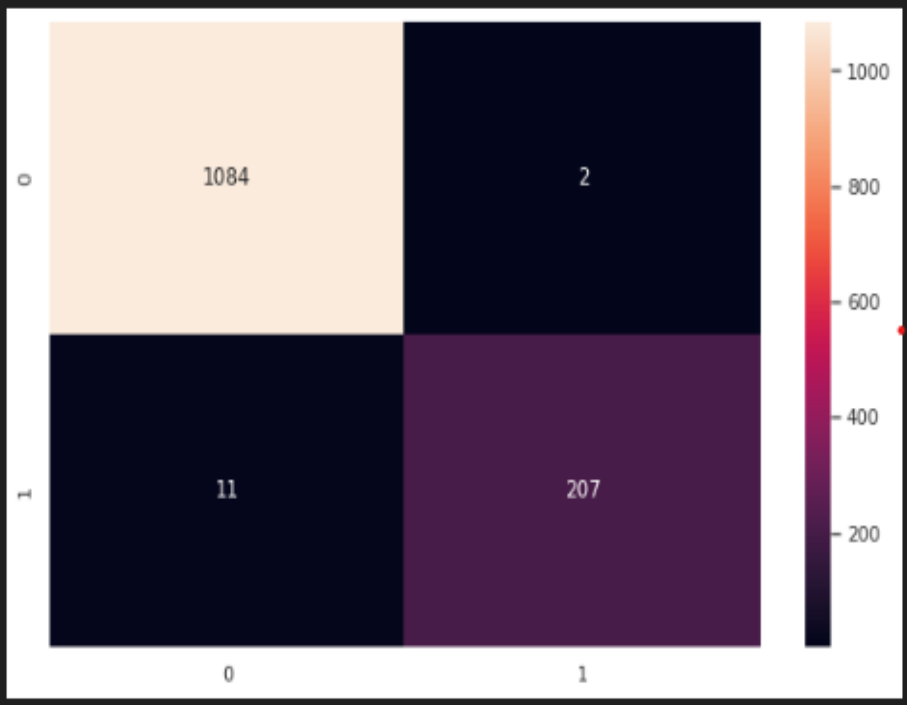
```
Accuracy : 0.9118098159509203

------------ Classification Report -----------
              precision    recall  f1-score   support

           0       0.91      1.00      0.95      1086
           1       0.97      0.49      0.65       218

    accuracy                           0.91      1304
   macro avg       0.94      0.74      0.80      1304
weighted avg       0.92      0.91      0.90      1304

------------ Confusion Matrix --------------
```



SVM

clf_svm = SVC ()
clf_svm.fit(X_train, y_train)
performance_eval(clf_svm,X_test)

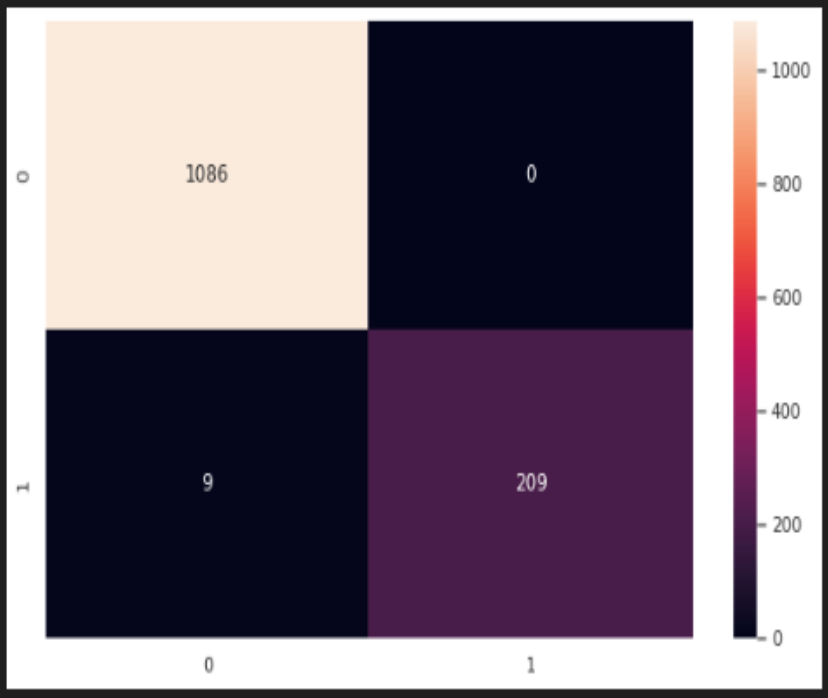
```
Accuracy : 0.9900306748466258

   ------------ Classification Report -----------
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1086
           1       0.99      0.95      0.97       218

    accuracy                           0.99      1304
   macro avg       0.99      0.97      0.98      1304
weighted avg       0.99      0.99      0.99      1304

   ------------ Confusion Matrix --------------
```



Random Forest

clf_rf = RandomForestClassifier()
clf_rf.fit(X_train, y_train)
performance_eval(clf_rf,X_test)

```
Accuracy : 0.9930981595092024

------------ Classification Report -----------
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      1086
           1       1.00      0.96      0.98       218

    accuracy                           0.99      1304
   macro avg       1.00      0.98      0.99      1304
weighted avg       0.99      0.99      0.99      1304

------------ Confusion Matrix --------------
```



```
voting_clf = VotingClassifier(estimators=[('LogReg', clf_lr),
                                          ('RF', clf_rf),
                                          ('AdaBoost',clf_adb),
                                          ('KNN',clf_knn),
                                          ('NB',clf_nb),
                                          ('DT',clf_dt)
                                         ],
                              voting='hard'
                             )
voting_clf.fit(X_train, y_train)
performance_eval(voting_clf,X_test)

import pickle
filename = 'CV_BestModel.sav'
pickle.dump(voting_clf, open (filename, 'wb'))

text = "I am / exhauted :) and restless."

clean_text = text_cleaner(text)
clean_text

loaded_model = pickle.load(open('CV_BestModel.sav', 'rb'))
single_prediction =
loaded_model.predict(vect.transform([clean_text]).toarray()) [0]

output = {0:"No Anxiety/Depression",1:"Anxiety/Depression"}
print(output[single_prediction])
```

ML Models Accuracy Visualization.ipynb

```
import pandas as pd
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import re
import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import HashingVectorizer
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC

df = pd.read_excel('dataset.xlsx')
df = df[:3000]

df.label.value_counts()
df.shape
df = df.sample(frac = 1)
df.head()
df.isnull().sum()
df.dropna()
sns.countplot(df.label)
```

```
import nltk
from nltk.corpus import stopwords
fromnltk.stemimportWordNetLemmatizer
importspacy
lm = WordNetLemmatizer()
nltk.download('wordnet')

deftext_transformation(col):
    corpus = []
    fortokenincol:
        alphabet = re.sub('[^a-zA-Z]',' ',str(token))
        alphabet = alphabet.lower()
        alphabet = alphabet.split()
        a_lemmas = [lm.lemmatize(word)
forwordinalphabetifwordnotinset(stopwords.words('english'))]
        corpus.append(' '. join(str(x) forxina_lemmas))
    returncorpus

df['cleaned'] = text_transformation(df.text)

X = df.cleaned
y = df.label.astype(int)

vect = TfidfVectorizer(max_features = 20000 , lowercase=False, ngram_range=
(1,2),use_idf = True)

X_tfidf =vect.fit_transform(X).toarray()

fromsklearn.model_selectionimporttrain_test_split
X_train,X_test,y_train,y_test = train_test_split(X_tfidf,y,stratify = y,test_size =
0.2,random_state = 1)

fromsklearn.metricsimportclassification_report,confusion_matrix,
accuracy_score

from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
```

```
lr_model = LogisticRegression(class_weight='balanced', max_iter=400)
dt_model = DecisionTreeClassifier(class_weight='balanced')
rf_model = RandomForestClassifier(class_weight='balanced')
mnb_model = MultinomialNB()
svm_model = SVC (class_weight='balanced')
knn_model = KNeighborsClassifier()
ada_model = AdaBoostClassifier()
model_names = ['Logistic Regression','DecisionTree','RandomForest','Naive
Bayes', 'SVM', 'KNN', 'AdaBoost']
ml_models = [lr_model,dt_model,rf_model,mnb_model, svm_model,
knn_model, ada_model]

trained_ml_models = []
defmodel_train(model, train_data, train_labels):
    mf = model.fit(train_data,train_labels)
    trained_ml_models.append(mf)
# trained_ml_models = []
foriinml_models:
    model_train(i,X_train, y_train)
#     trained_ml_models.append(tt)

fromsklearn.metricsimportaccuracy_score,precision_score,recall_score,f1_score
,roc_auc_score

pr_list = []
re_list = []
f1_list = []

defprint_results(md, x_test, y_test, name):
    #print("\n\nClassifier: ", name)
    pred_y = md.predict(x_test)
    true, pred = y_test, pred_y

    pr_list.append((round(precision_score(y_test, pred_y, average='weighted'),4)
*100))
    re_list.append((round(recall_score(y_test, pred_y, average='weighted'),4)
*100))
```

```
    f1_list.append((round(f1_score(y_test, pred_y, average='weighted'),4)*100))

foriinrange(len(trained_ml_models)):
    md = trained_ml_models[i]
    name = model_names[i]
    print_results(md, X_test, y_test, name)

performance_matrix = pd.DataFrame({'Precision':pr_list,
                                   'Recall':re_list,'F1 Score': f1_list},
index =model_names)
```

performance_matrix

Model	Precision	Recall	F1 Score
Logistic Regression	94.73	94.67	94.51
Decision Tree	98.50	98.50	98.49
Random Forest	96.86	96.83	96.78
Naive Bayes	88.41	86.67	84.79
SVM	92.29	91.83	91.32
KNN	79.44	77.33	69.54
AdaBoost	97.34	97.33	97.30

```
data_matrix = pd.DataFrame({
    'Model': model_names,'Precision':pr_list,
                        'Recall':re_list,
                        'F1 Score': f1_list,
                      },
                    )
```

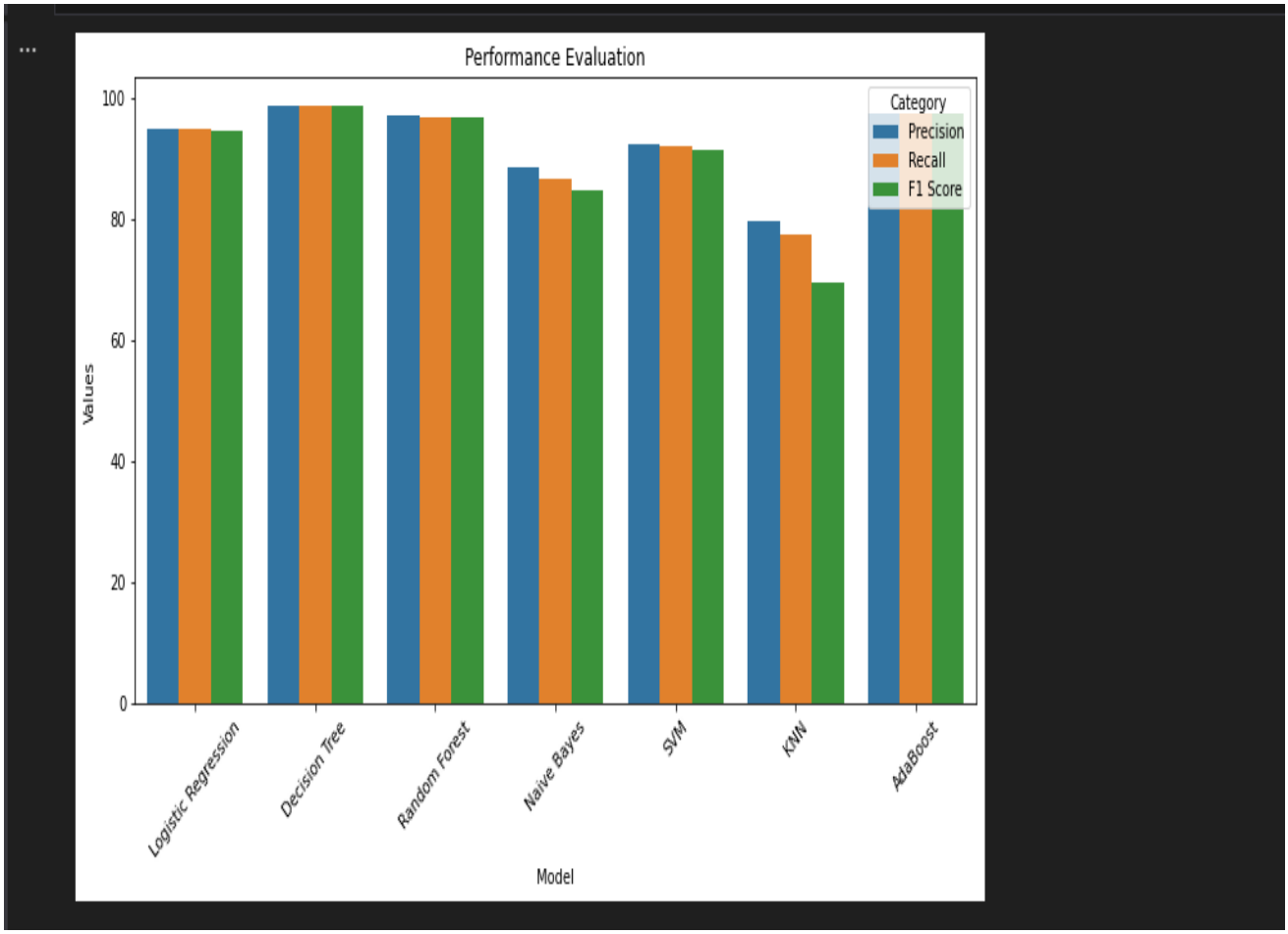
data_matrix

	Model	Precision	Recall	F1 Score
0	Logistic Regression	94.73	94.67	94.51
1	Decision Tree	98.50	98.50	98.49
2	Random Forest	96.86	96.83	96.78
3	Naive Bayes	88.41	86.67	84.79
4	SVM	92.29	91.83	91.32
5	KNN	79.44	77.33	69.54
6	AdaBoost	97.34	97.33	97.30

```
df_1 = pd.melt(data_matrix, id_vars="Model", var_name="Category",
value_name="Values")
plt.figure(figsize= (12, 6))
ax = plt.subplot()

sns.barplot(data=df_1,x='Model', y='Values' ,hue='Category')
ax.set_xlabel('Model')
ax.set_title('Performance Evaluation')

ax.xaxis.set_ticklabels(model_names, rotation=45);
plt.show()

df_1 = pd.melt(data_matrix, id_vars="Model", var_name="Category",
value_name="Values")
plt.figure(figsize= (12, 6))
ax = plt.subplot()

sns.barplot(data=df_1,x='Model', y='Values' ,hue='Category')
ax.set_xlabel('Model')
```

ax.set_title('Performance Evaluation')

ax.xaxis.set_ticklabels(model_names, rotation=45);
plt.show()



PREDICTION.PY

```
import joblib
from sklearn.feature_extraction.text import CountVectorizer

def xgb_predict(sentence):
  data_list = [sentence]
  xg = joblib.load("xgb_model.sav")
  cv = joblib.load("vectorizer.pkl")
  cv2 =
CountVectorizer(vocabulary=cv.vocabulary_,lowercase=True,stop_words='engl
ish',ngram_range = (1,1))
  X_test1= cv2.transform(data_list)
  predicted2= xg.predict(X_test1)
  return predicted2


def lr_predict(sentence):
  data_list = [sentence]
  lr = joblib.load("logistic_model.sav")
  cv = joblib.load("vectorizer.pkl")
  cv2 =
CountVectorizer(vocabulary=cv.vocabulary_,lowercase=True,stop_words='engl
ish',ngram_range = (1,1))
  X_test1= cv2.transform(data_list)
  predicted2= lr.predict(X_test1)
  return predicted2

def xgb_suicide(sentence):
  data_list = [sentence]
  xg = joblib.load("suicide_xgb_model.sav")
  cv = joblib.load("suicide_vectorizer.pkl")
  cv2 =
CountVectorizer(vocabulary=cv.vocabulary_,lowercase=True,stop_words='engl
ish',ngram_range = (1,1))
  X_test1= cv2.transform(data_list)
  predicted2= xg.predict(X_test1)
  return predicted2
```

```
def lr_suicide(sentence):
  data_list = [sentence]
  lr = joblib.load("suicide_logistic_model.sav")
  cv = joblib.load("suicide_vectorizer.pkl")
  cv2 =
CountVectorizer(vocabulary=cv.vocabulary_,lowercase=True,stop_words='engl
ish',ngram_range = (1,1))
  X_test1= cv2.transform(data_list)
  predicted2= lr.predict(X_test1)
  return predicted2
```

Danger_word.py

```
def GetDangerWords():
    words_file = open("concerning_words.txt")
    danger_words = []
    for word in words_file:
        danger_words.append(word.rstrip())
    return danger_words
```

App.py

```
import streamlit as st
import pandas as pd
import numpy as np
from prediction import xgb_predict, lr_predict, xgb_suicide
import nltk
from nltk import sent_tokenize, word_tokenize
import pdfplumber
import danger_words
from fpdf import FPDF

nltk.download('punkt')

# Function to convert a string to a PDF file
def StringToPDF(string):
    pdf = FPDF()
    pdf.add_page()
    # Ensure the font file 'DejaVuSansCondensed.ttf' is in the same directory
    pdf.add_font('DejaVu', '', 'DejaVuSansCondensed.ttf', uni=True)
    pdf.add_font('DejaVu', 'B', 'DejaVuSansCondensed-Bold.ttf', uni=True)
    pdf.set_font('DejaVu', size=14)
    pdf.multi_cell(190, 10, txt=string)
    return bytes(pdf.output(dest='S').encode('latin1'))

def WriteToFile(flaglist, title):
    string = "\n*" + title + "*\n"
    for i, flag in enumerate(flaglist, start=1):
        string += str(i) + ". " + flag.replace("\n", " ") + "\n"
    if not flaglist:
        string += "No red flags found!"
    string += "\n"
    return string

# Function to analyze and display flags from text input
def ShowFlags(flaglist):
```

```
    count = 0
    flagcount = len(flaglist)
    if flagcount == 0:
        st.write("No red flags found!")
    else:
        with st.expander("Found " + str(flagcount) + " red flag/s:", True):
            for flag in flaglist:
                count += 1
                st.write(str(count) + ". " + flag)
            st.write("\n")

# Function to analyze text for red flags
def Display(essay, progress=None, filename=None):
    if essay:
        sentences = sent_tokenize(essay)
        xgflags = []
        lrflags = []
        xgsflags = []
        wrflags = []

        concerning_words = danger_words.GetDangerWords()

        loading = st.progress(0, text="Loading...")
        for i, sentence in enumerate(sentences, start=1):
            loading.progress(i / len(sentences), text=f"Analyzing sentence
{i}/{len(sentences)}...")
            if xgb_predict(sentence) == 1:
                xgflags.append(sentence)
            if lr_predict(sentence) == 1:
                lrflags.append(sentence)
            if xgb_suicide(sentence) == 1:
                xgsflags.append(sentence)
            if any(word in concerning_words for word in
word_tokenize(sentence.lower())):
                wrflags.append(sentence)

        loading.progress(1, text="Complete!")
```

```
        with col1:
            st.write("### Concerning Words Test")
            ShowFlags(wrflags)
        with col2:
            st.write("### Potential Struggles Test")
            ShowFlags(lrflags)
        with col3:
            st.write("### Depressive Thoughts Test")
            ShowFlags(xgflags)
        with col4:
            st.write("### Destructive Thoughts Test")
            ShowFlags(xgsflags)

        if filename:
            string = ("Results for " + filename + ":\n").upper()
        else:
            string = "Results:\n"
        string += WriteToFile(xgsflags, "Destructive Thoughts")
        string += WriteToFile(xgflags, "Depressive Thoughts")
        string += WriteToFile(lrflags, "Potential Struggles")
        string += WriteToFile(wrflags, "Concerning Words")
        return string

# Function to extract text from PDF file
def GetPDFText(uploaded_file):
    data = ""
    try:
        with pdfplumber.open(uploaded_file) as current_pdf:
            for page in current_pdf.pages:
                data += page.extract_text()
    except Exception as e:
        st.error(f"Error extracting text from PDF: {e}")
    return data

# Function to refresh the app
def Refresh():
    st.stop()
```

```
st.title("Welcome to [Mental Stress Predictor]")
st.subheader("Text-based Early Distress Detector for Youth",
anchor="welcome-to-Mental-Stress-Predictor")
st.caption("In the sidebar, enter a sufficient amount of text* that is reflective of
a person's thoughts: essays, reflections, and chat conversations work best.
Mental Stress Predictor will use artificial intelligence to display sentences that
may be a cause for concern.\n:red[Mental Stress Predictor is not meant to be
used as a diagnostic tool] - it is designed to give you a general idea of whether
someone in your school or workplace might need more emotional support.")
st.write("\n")

col1, col2, col3, col4 = st.columns(4)

with st.sidebar:
    st.title("Run an essay/conversation through our text analyzer!")
    essay = st.text_area("Copy-paste text here:")
    if st.button("Submit Text"):
        string = Display(essay)
        st.download_button("Download Report", StringToPDF(string),
file_name="Report.pdf")
    uploaded_file = st.file_uploader('Or, upload a PDF file:', type="pdf")
    if uploaded_file and st.button("Submit File"):
        string = Display(GetPDFText(uploaded_file), "1/1", uploaded_file.name)
        st.download_button("Download Report", StringToPDF(string),
file_name="Report.pdf")
```

OUTPUT

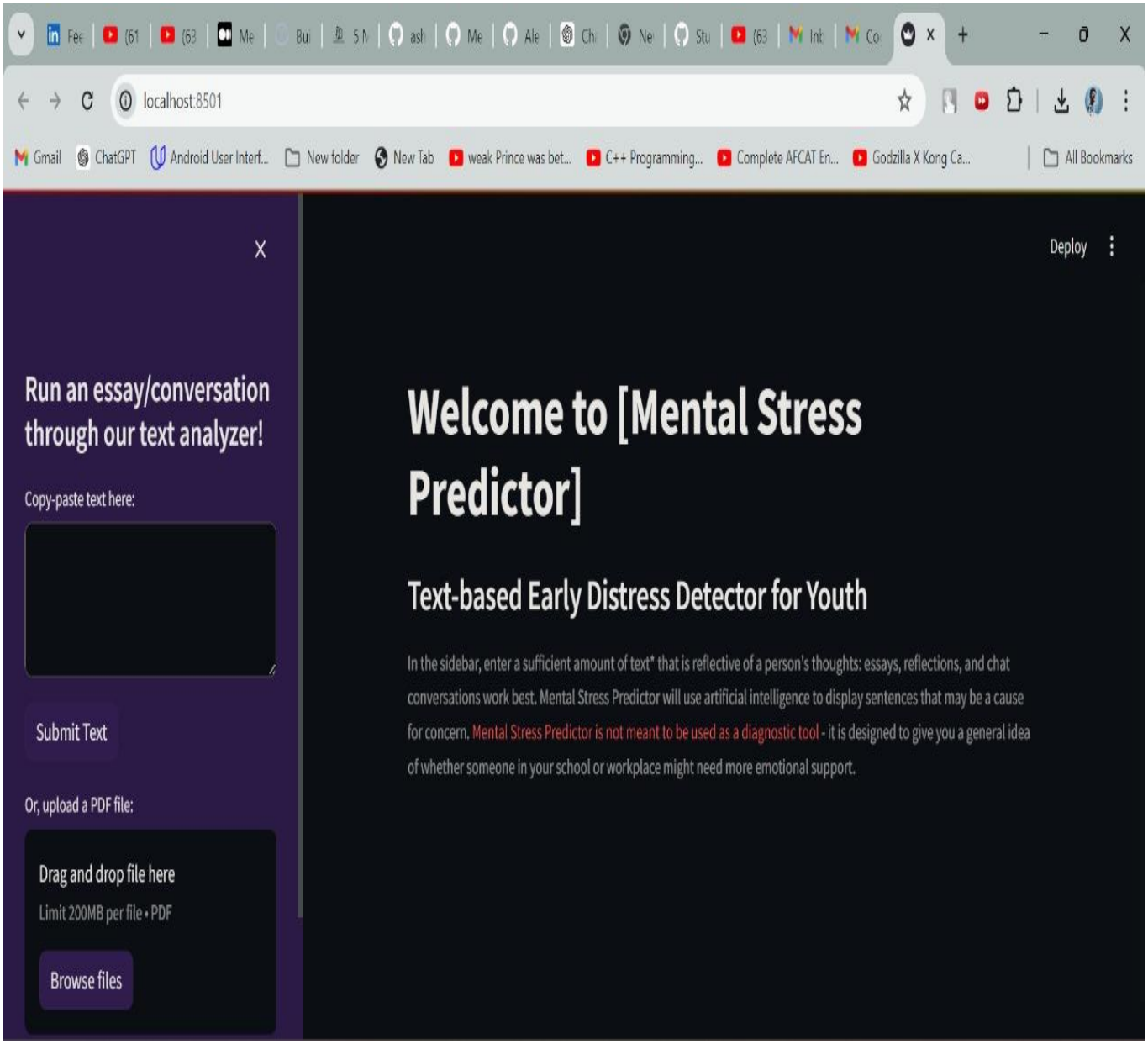


Fig.1 Sample page Layout

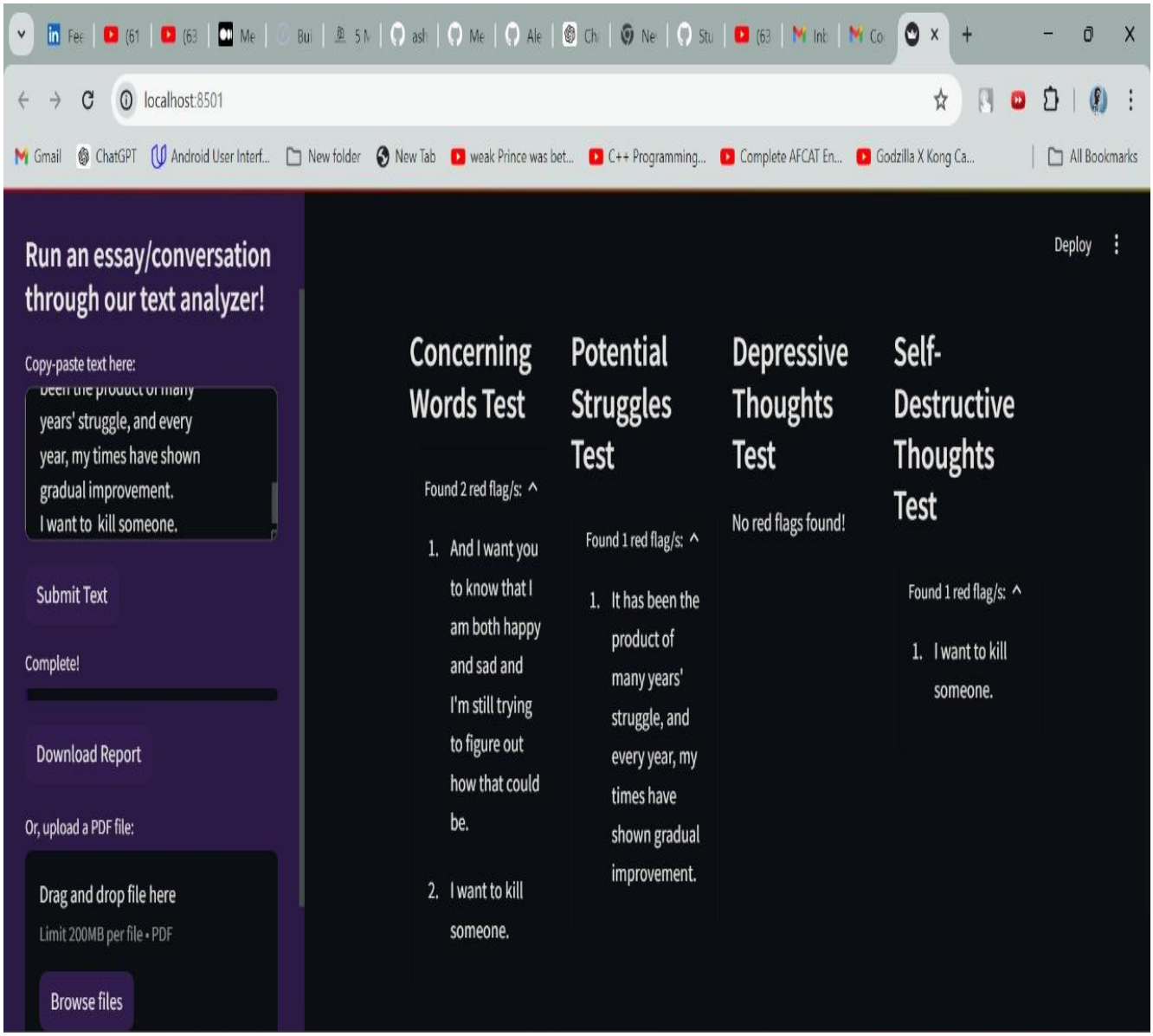


Fig2. Output After filling Text area

Input Text -- So, this is my life. And I want you to know that I am both happy and sad and I'm still trying to figure out how that could be.

The longer and more carefully we look at a funny story, the sadder it becomes.

Look at my success. I didn'tachieve it overnight. It hasbeen the product of many years' struggle, and everyyear, my times have shown

gradual improvement.

I wana also like to kill someone.

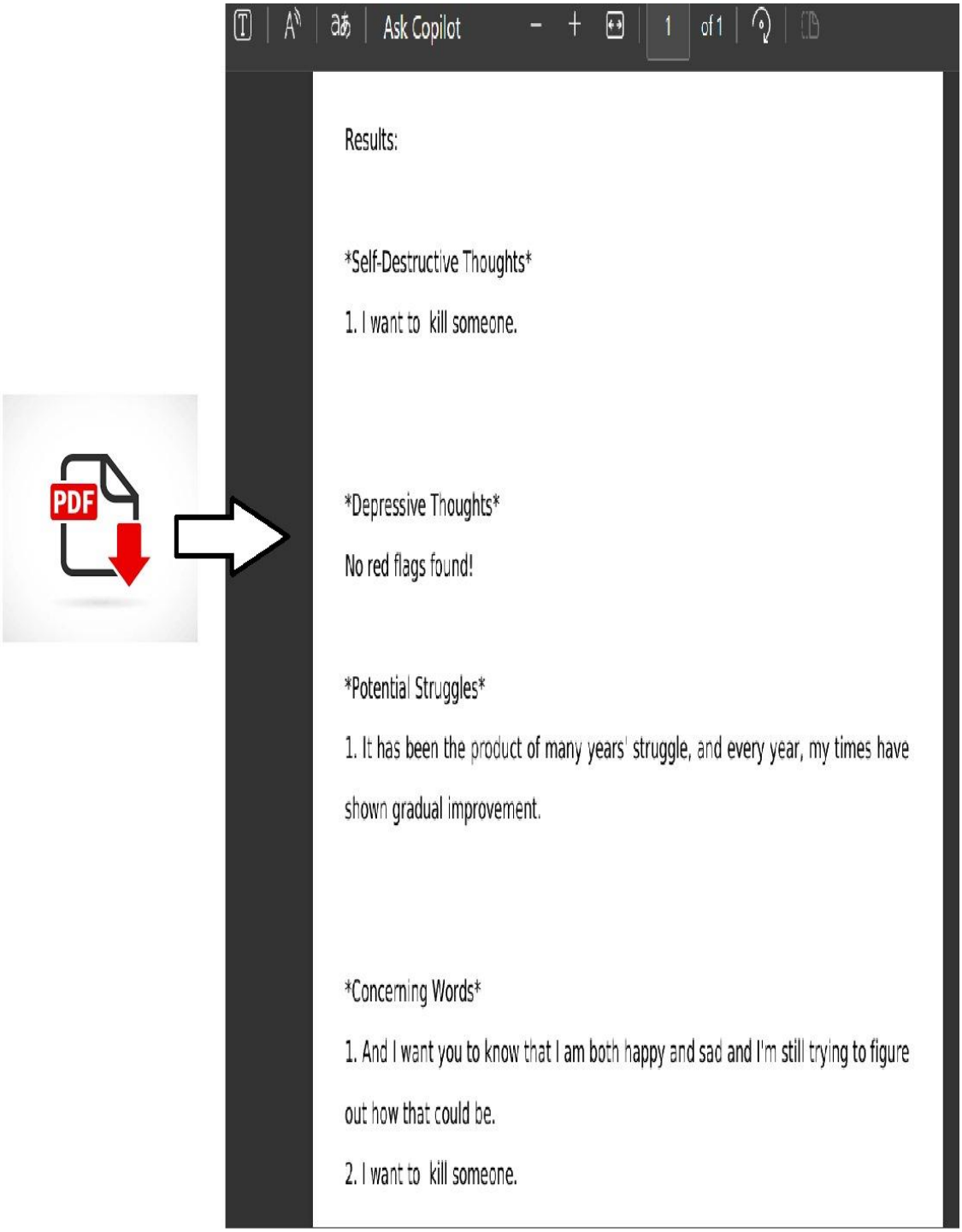


Fig 3. PDF Report (after submitting the textual data)

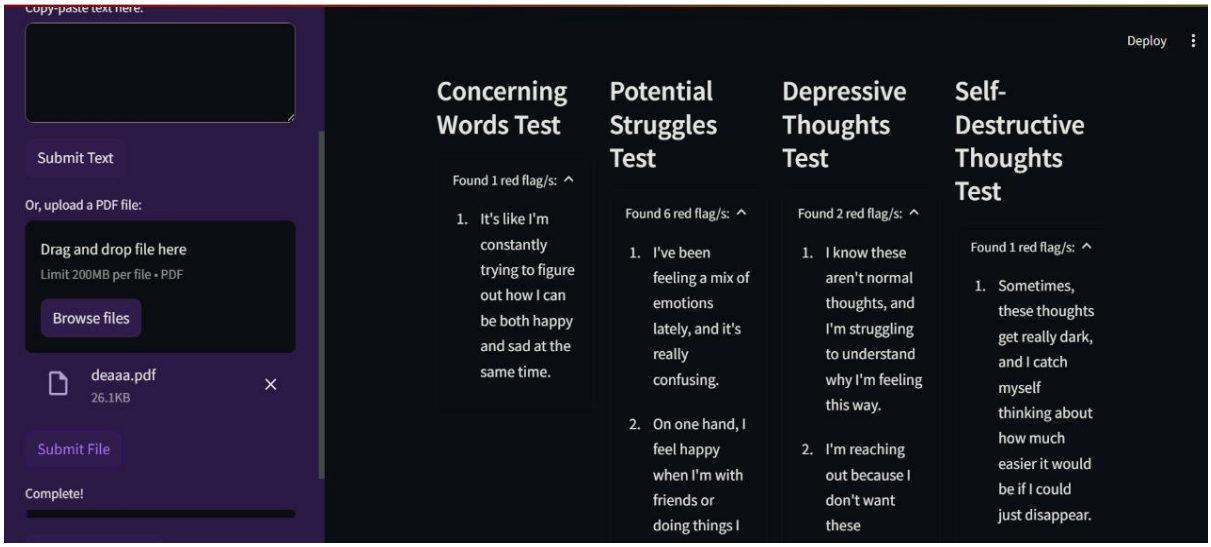


Fig 4.1 Output after PDF upload

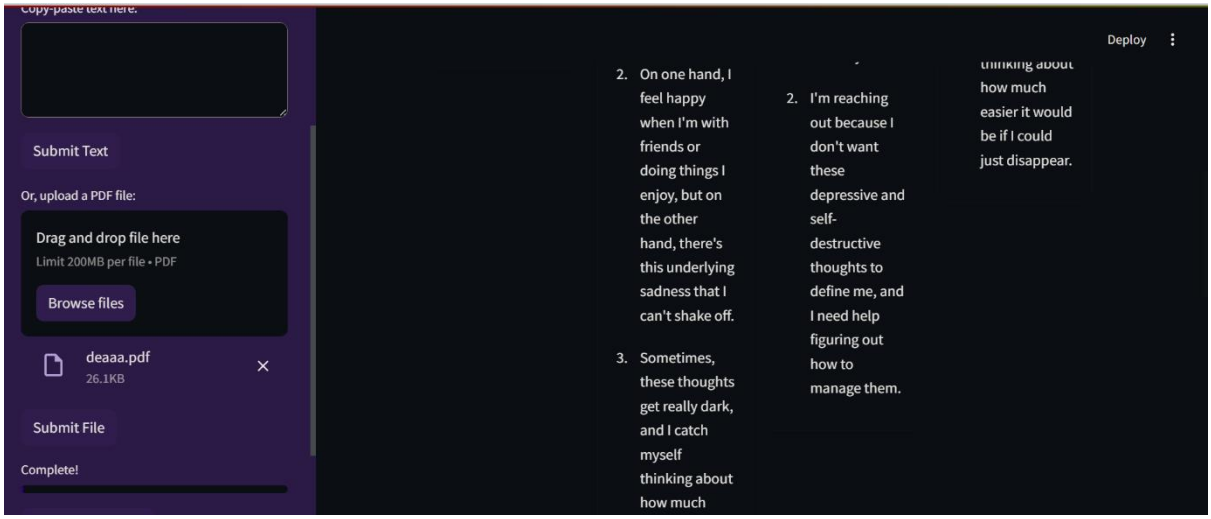


Fig 4.2 Output after PDF upload

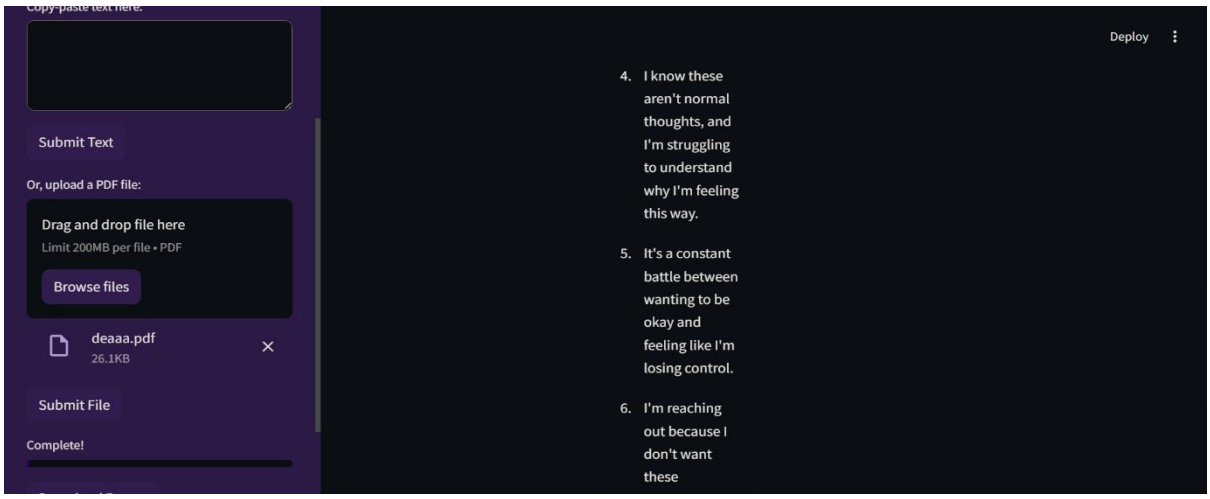


Fig 4.3 Output after PDF upload

RESULTS FOR DEAAA.PDF:

Self-Destructive Thoughts

1. Sometimes, these thoughts get really dark, and I catch myself thinking about how much easier it would be if I could just disappear.

Depressive Thoughts

1. I know these aren't normal thoughts, and I'm struggling to understand why I'm feeling this way.
2. I'm reaching out because I don't want these depressive and self-destructive thoughts to define me, and I need help figuring out how to manage them.

Potential Struggles

1. I've been feeling a mix of emotions lately, and it's really confusing.
2. On one hand, I feel happy when I'm with friends or doing things I enjoy, but on the other hand, there's this underlying sadness that I can't shake off.
3. Sometimes, these thoughts get really dark, and I catch myself thinking about how much easier it would be if I could just disappear.
4. I know these aren't normal thoughts, and I'm struggling to understand why I'm feeling this way.
5. It's a constant battle between wanting to be okay and feeling like I'm losing control.
6. I'm reaching out because I don't want these depressive and self-destructive thoughts to define me, and I need help figuring out how to manage them.

Concerning Words

1. It's like I'm constantly trying to figure out how I can be both happy and sad at the same time.

Fig 5 report result after output download

REFERENCES:

1. https://www.who.int/news-room/fact- sheets/detail/mental-disorders
2. https://www.who.int/news-room/fact-sheets/detail/depression
3. Reshma Radheshamjee Baheti, Supriya Kinariwala. "Detection and Analysis of Stress using Machine Learning Techniques." Issue-1, October 2019 [CrossRef]
4. Disha Sharma, Nitika Kapoor & Dr. Sandeep Singh Kang. "Stress prediction of students using machine learning." Issue 3, Jun 2020. [CrossRef]
5. Mike Thelwall TensiStrenGth, "Stress and relaxation magnitude detection for social media text", 2017
6. Sakshi Yadav detecting presence of mental illness using nlp sentiment analysis, 2022
7. Thelwall M., Buckley, Paltoglou, Cai, Kappas, "Sentiment strength detection in short informal text", 2010.
8. Umar Rashid, Muhammad Waseem Iqbal, Muhammad Akmal Skiandar. "Emotion Detection of Contextual Text using Deep learning." 2020. [CrossRef]
9. Information Zhentao Xu, Veronica Perez-Rosas, Rada Mihalcea, Inferring Social Media Users' Mental Health Status from Multimodal, 2020 [CrossRef]
10. Varun Sundaram, Saad Ahmed, Shaik Abdul Muqtadeer. "Emotion Analysis in Text using TF- IDF" , 2021[CrossRef]
11. De Choudhury, M., Gamon, M., Counts, S., and Horvitz, E. Predicting depression via social media. (2013) [CrossRef]
12. Moreno, M. A., Jelenchick, L. A., Egan, K. G., Cox, E., Young, H., Gannon, K. E., and Becker, T. Feeling bad on facebook, (2011).
13. De Choudhury, E. Dredze, M., Coppersmith,G., and Kumar, M. (2016). Discovering shifts to suicidal ideation from mental health content in social media.
14. Park, M., Cha, C., and Cha, M. (2012). Depressive moods of users portrayed in twitter.