Evaluating Psychometric Features and Contextual Embeddings Potential for Mental Disorder Classification in Short Biomedical Texts

submitted in partial fulfilment of the requirements
for the degree of

Master of Science

in

Mathematics

by

Suryam Gupta

(Admission No.: I19MA038)

under the supervision of

Dr. Sandra Mitrović

Full-Time Teacher and Postdoctoral Researcher, IDSIA, USI/SUPSI, Switzerland

&

Dr. Amit Sharma

Assistant Professor, Sardar Vallabhbhai National Institute of Technology, Surat, India

Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
Surat-395007, Gujarat, India.

Department of Mathematics,
Sardar Vallabhbhai National Institute of Technology,
(An Institute of National Importance, NIT Act 2007)
Surat-395 007, Gujarat, INDIA.

DECLARATION

I hereby declare that the dissertation entitled **"Evaluating Psychometric Features and Contextual Embeddings Potential for Mental Disorder Classification in Short Biomedical Texts"** is a genuine record of research work carried out by me and no part of this dissertation has been submitted to any university or institution for the award of any degree or diploma.

Suryam Gupta
(I19MA038)
Department of Mathematics,
Sardar Vallabhbhai National Institute of Technology,
Surat - 395007.

Date: June 25, 2024
Place: Surat

Department of Mathematics,
Sardar Vallabhbhai National Institute of Technology,
(An Institute of National Importance, NIT Act 2007)
Surat-395 007, Gujarat, INDIA.

Date: June 25, 2024

CERTIFICATE

This is to certify that the dissertation entitled **"Evaluating Psychometric Features and Contextual Embeddings Potential for Mental Disorder Classification in Short Biomedical Texts"** is based on a part of research work carried out by Mr. Suryam Gupta under our guidance and supervision.

Dr. Sandra Mitrović
(External Faculty)
Full-Time Teacher and Postdoctoral Researcher
IDSIA (Dalle Molle Institute for Artificial Intelligence)
USI/SUPSI (University of Southern Switzerland and University of Applied Sciences and Arts of Southern Switzerland), Switzerland

Dr. Amit Sharma
(Internal Faculty)
Assistant Professor
Department of Mathematics
Sardar Vallabhbhai National Institute of Technology, Surat, India.

Department of Mathematics,
Sardar Vallabhbhai National Institute of Technology,
(An Institute of National Importance, NIT Act 2007)
Surat-395 007, Gujarat, INDIA.

Approval Sheet

The dissertation entitled **"Evaluating Psychometric Features and Contextual Embeddings Potential for Mental Disorder Classification in Short Biomedical Texts"** by Mr. Suryam Gupta is approved for the degree of Master of Science in Mathematics.

Dr. Amit Sharma
(Supervisor)

Prof. A.K. Shukla
(Examiner I)

Dr. Shivam Bajpeyi
(Examiner II)

Date: June 25, 2024
Place: Surat

ACKNOWLEDGEMENT

I express my sincere gratitude to **Dr. Sandra Mitrović** for her unwavering guidance, patience, and support throughout the semester and the dissertation writing process. Her assistance has been extremely helpful in my personal and professional growth. I also thank **Dr. Amit Sharma** for encouraging me to work hard on this dissertation and providing advice when needed.

I am deeply grateful to my parents for their unwavering love, support, and encouragement throughout my academic journey. Their constant belief in my abilities and their willingness to provide both emotional and financial support has been invaluable.

I extend heartfelt gratitude to the seniors whom I met during my college years. Among the seniors, I would like to especially thank **Mr. Ashwin Verma, Mr. Piyush Sharma, and Mr. Harsh Chauhan** for their guidance in both academic and non-academic matters. Their support has been invaluable, and I am truly grateful for their mentorship.

I have also been fortunate to have batchmates like **Kaushik Sanghani, Priyanshi Chandra, Mridul Sehgal, Yogesh Goyal, Ajay Khasiya, Ayushi Singh, and Deepshikha Rathore**, who have made the coursework seem a lot easier and more enjoyable through their companionship and support. Furthermore, I express deep gratitude to the people I met throughout my 5 years at SVNIT. The friendships we formed and the memories we created together have brought immense joy and enrichment to my college experience. Among them, I am highly grateful to have met **Shreyan Goswami, Abhishek Mondal, Hershini Gadaria, Sreejita Das, Mukul Raj Mishra, Shivam Rajpoot, and Sarvesh Kumar**. Their unwavering belief in me, often more than I believed in myself, has been a tremendous source of encouragement. They have consistently inspired me to strive for self-improvement and have been a constant source of motivation throughout my college journey.

I would also like to thank **Dr. Jayesh M. Dhodiya**, Head of the Department of Mathematics, and the entire department faculty for their support and encouragement during my studies.

At last, I thank all those who have helped me directly or indirectly in this project's success.

Suryam Gupta

ABSTRACT

With the significant rise in mental health awareness, detecting mental illnesses has become a major concern. Psychiatrists often struggle to identify mental health issues in patients due to the complex nature of each disorder, making it challenging to provide timely and appropriate treatment. However, the widespread use of social media in daily life offers an environment that could provide valuable insights into a patient's mental health condition. It provides an opportunity to leverage user-generated content for mental health diagnosis, and understanding how individuals express mental health concerns online can enhance early diagnosis and treatment strategies.

This study explores the classifying capabilities of psychometric features and contextual embeddings (both separately and jointly) in short biomedical literature. It incorporates the use of machine learning and deep learning models to classify mental health conditions such as anxiety, panic, and depression using social media data from Reddit. The dataset comprised 5,441 posts from relevant subreddits, and features were extracted including contextual embeddings (SBERT), Linguistic Inquiry and Word Count (LIWC features), emotion detection, and emotional intensity. Logistic Regression (LR) and Extreme Gradient Boosting (XGB) models were employed, with the LR model achieving the highest F1 score of 81.9%. A multi-layered deep learning model was also tested, yielding an F1 score of 80.7%.

Index Terms: Mental Health, Social Media, Machine Learning, Natural Language Processing, Logistic Regression, Extreme Gradient Boosting, Deep Learning, Contextual Embeddings, Emotion Detection, Emotional Intensity, Linguistic Inquiry and Word Count (LIWC), Bidirectional Encoder Representations from Transformers (BERT)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Mental health includes our emotional, psychological, and social well-being, significantly influencing our thoughts, feelings, and behaviors. It's crucial for overall health, affecting daily life, physical health, and relationships. Addressing mental health is vital for fostering resilient individuals and supportive communities [1]. In the past decade, social media has increasingly been used to express personal thoughts, emotions, and ideas. In healthcare, online posts have been utilized to detect disease outbreaks, identify smoking patterns, and recognize adverse drug reactions. A promising application is the automatic detection of mental health issues, a burgeoning field attracting researchers in Natural Language Processing (NLP). Resources from Twitter, Facebook, blogs, online forums, Reddit, and Quora are used to detect various mental health issues, such as anxiety, depression, suicidal ideation, and eating disorders [2].

The current literature on tracking mental disorders in social media underscores the importance of this topic but is biased towards specific disorders like depression [3] [4], anorexia [5], bipolar disorder, ADHD, PTSD [6], and suicide [7]. Anxiety and panic disorders are particularly understudied. Additionally, occasional and less frequent mental health episodes are often overlooked in current studies.

Anxiety is characterized by the activation of the behavioral inhibition system (BIS) and includes conditions marked by excessive fear, emotional responses to real or perceived threats, and worry about future threats [8]. Common forms of anxiety include generalized anxiety, social anxiety, and health anxiety. Panic attacks are defined by the Diagnostic and Statistical Manual of Mental Disorders as "a sudden surge of intense fear or intense discomfort that reaches a peak within minutes." [9] Symptoms may include palpitations, accelerated heart rate, sweating, trembling, smothering, and chest pain. The recurrence and worsening of panic attacks are often due to conditioned responses becoming fixed in the mind, capable of actualizing the fear of death [10]. Depression is a common, debilitating, and potentially fatal disorder affecting over 300 million people worldwide. It is the leading contributor to global disability according to the WHO [11]. Depression can negatively impact a person's thinking, feeling, and actions, and has increased over the past decade. Adolescents with major depressive disorder are significantly more likely to commit suicide. Lack of appropriate treatment can lead to disability, psychotic episodes, self-harm, and suicide.

Despite the seriousness of mental disorders, stigma persists, suggesting weakness and often leading to social exclusion. Studies show that while people recognize mental health issues as a serious problem, they often doubt their treatability. This stigma may prevent individuals from seeking professional help, with 75% - 85% of people not receiving proper care [12].

Given the high prevalence of these disorders, there is an urgent need for scalable mental health detection tools accessible to large populations. The frequent discussions of mental health on social media underscore the potential for developing these tools on various communication platforms.

1.2 Significance of Research

This research carries significant potential for individuals, governments, and countries by leveraging the latest Natural Language Processing (NLP) techniques to enhance mental health diagnosis and treatment. It will contribute to a deeper understanding of how individuals express mental health concerns on social media, revealing common themes, emotions, and experiences. Quicker and more accurate diagnoses of mental disorders can lead to improved treatment outcomes and faster recoveries. By providing healthcare professionals with data-driven insights, this research empowers them to make more informed decisions, ultimately enhancing the quality of care. Explainable models developed through this research build public trust in mental health diagnosis and treatment, fostering greater acceptance and understanding. Additionally, advancements in this field raise awareness of the importance of mental health, promoting help-seeking behaviors and reducing stigma. The innovations proposed can offer affordable and cost-effective healthcare solutions, particularly in low-resource settings, ensuring accessibility for all. Furthermore, this research aims to support public health institutions by making resources readily accessible for addressing mental health disorders, thereby contributing to the development of resilient individuals and supportive communities. Overall, the study's implications extend beyond academic contributions, influencing public health strategies and interventions aimed at improving mental well-being on a global scale.

1.3 Problem Statement and Objectives

An automated system capable of identifying elevated mental disorder scores in users could enable targeted interventions, including comprehensive assessments and the provision of necessary resources, support, and treatment. Existing research has either explored the correlation between social media usage and mental illness or focused on detecting mental illness through the analysis of user-generated content. This review emphasizes the latter, aiming to predict mental illness using social media data.

To the best of our knowledge, current literature lacks computational approaches for understanding panic and anxiety in social media textual resources (apart from the works of [13] and [14]). Moreover, no existing study has considered anxiety, panic, and depression data simultaneously to classify these conditions. To address this gap, we compiled a dataset of 5,441 user posts from the popular social media platform Reddit, using various subreddits to extract posts related to anxiety, panic, and depression.

The following is an example of an Anxiety, Panic, and Depression text collected from Reddit.

Anxiety

> *"i wake up with this dreadful feeling every morning. it's nearly crippling, i feel exhausted and scared. it makes starting every day so difficult and makes me wanna just stay home and sleep rather than go to school. i wish i knew a way to stop it. i want to start medication for my anxiety but i'm not sure how soon that'll be put into action. does anyone have a similar situation and possibly a way to cope?"*

Panic

> *"it is hard not to feel valid when all my panic attacks are not even visible to onlookers. people do not understand the turmoil i go through on a daily basis cause they cannot see it. some of my symptoms during panic attacks include racing heart, shortness of breath, tight chest, depersonalization, numbness/tingling, muscle twitches (rare occasions), and dry mouth. sometimes i feel i may die due to them. i once had to go to the er because i was so terrified i was dying. they feel absolutely horrible, and have really impacted my life on a daily basis. they happen out of the blue, and i wake up everyday scared of when the next one will happen. but i feel like nobody takes my condition seriously, since they cannot even see when im panicking. like maybe im just overreacting. does anybody else deal with invisible panic attacks? :("*

Depression

> *"i feel so lonely. my friends are too busy with their own lives. i dont want to bother them. yet they're persistent on me staying alive. i feel so drained, useless, exhausted, alone, worthless and everything else. no professional wants to help me, even after attempting. big sigh."*

We begin by employing methods to extract several features from the texts, including:

- Contextual Embeddings
- LIWC Features
- Emotion Detection
- Emotion Intensity

To evaluate the classification capabilities of these features, we build and compare three machine learning and deep learning models: Logistic Regression, Extreme Gradient Boosting (XGB), and a Multi-layered Deep Learning architecture. These models are then assessed using suitable evaluation metrics.

Chapter 2

Literature Review

2.1 Mental Disorder Classification

Recent advancements in machine learning (ML) and the growth of social media have enabled innovative methods for detecting mental disorders such as anxiety, depression, stress, and others. This literature review discusses three key studies that delve into the classification of various mental health conditions, focusing on the datasets used, features extracted, ML/DL models applied, and the results of their evaluations.

Kim et al. (2020) [15] focused on detecting a variety of mental illnesses through their proposed model, which effectively identified posts related to specific mental disorders such as depression, anxiety, bipolar disorder, borderline personality disorder (BPD), schizophrenia, and autism. The dataset included Reddit posts from six specific subreddits, with features extracted using TF-IDF vectorizer and word2vec embeddings. The study utilized six different binary classifiers and analyzed the features using XGBoost and convolutional neural networks (CNN). The CNN achieved a high accuracy rate of 96.96% for the r/autism class. Evaluation metrics demonstrated significant precision, recall, and F1 score for the CNN, highlighting its ability to recognize complex patterns associated with mental health conditions.

Ahmed et al. (2020) [16] developed a model to detect depression and anxiety in Bengali patients. The dataset was composed of annotated texts collected through two questionnaires designed by the Department of Psychology at the University of Dhaka. The proposed model applied five different AI algorithms: CNN, SVM, linear discriminant analysis, KNN, and linear regression on the anxiety and depression datasets. Based on various evaluation metrics (accuracy, recall, and precision), the CNN algorithm achieved the highest accuracy, with 96% for anxiety and 96.8% for depression.

Shen et al. (2017) [17] investigated depression detection by integrating multiple data modalities from social media, including text, images, and user behavior. The dataset consisted of well-labeled depression and non-depression data from Twitter. Six feature groups were extracted, covering clinical depression criteria and online behaviors on social media, including social network features, user profile features, visual features, emotional features, topic-level features, and domain-specific features. The study employed a multimodal depressive dictionary learning (MDL) model, significantly enhancing the model's robustness. Evaluation metrics such as accuracy, macro-averaged recall, macro-averaged precision, and macro-averaged F1-score were utilized. The MDL method achieved the highest performance with an 85% F1-score, indicating the effectiveness of combining a multimodal strategy with dictionary learning for depression detection.

2.2 Emotion Detection in Text

Emotion detection in text has become a pivotal research area in natural language processing (NLP), with numerous studies exploring various datasets and methodologies to identify and classify emotions accurately. This review focuses on three significant works in the field, examining the datasets used, the emotions considered, the machine learning and deep learning models applied, and the evaluation metrics employed.

Baziotis et al. (2018) [18] utilized the SemEval 2018 Task 1 competition dataset, which comprised tweets labeled with multiple emotions for their study. This dataset facilitated multi-label emotion classification across 11 emotions: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust. The authors proposed a deep learning model featuring a two-layer Bi-LSTM network enhanced with a multi-head self-attention mechanism. This architecture enabled the model to capture long-range dependencies and focus on crucial parts of the input text effectively. Despite their innovative approach, the researchers found that pre-training the Bi-LSTMs on the SemEval 2017 Task 4A dataset as a form of transfer learning did not outperform the model initialized randomly. This result underscored the complexity and challenge of enhancing model performance through transfer learning in this context.

In another significant study, Ragheb et al. (2019) [19] employed the SemEval 2019 Task 3 dataset, which included multi-party conversational data annotated with emotions. The study focused on multi-label emotion classification across four emotions: happy, sad, angry, and others. The authors introduced a deep learning model that combined transfer learning, self-attention mechanisms, and turn-based conversational modeling. The self-attention component was crucial for capturing significant contextual information, while the turn-based model tracked emotional dynamics across conversation turns. The model achieved an impressive F1-score of 0.7582, demonstrating the effectiveness of incorporating conversational context and attention mechanisms in emotion detection from textual data.

Huang et al. (2019) [20] also focused on the SemEval 2019 Task 3 dataset, which contained multi-party conversational data. Similar to Ragheb et al., they aimed at multi-label emotion classification across four emotions. They proposed a novel approach involving Hierarchical LSTMs for Contextual Emotion Detection (HRLCE), complemented by the use of large pre-trained language models like BERT. The hierarchical LSTM architecture was designed to model conversational context at both the utterance and speaker levels, capturing the nuances of conversation more effectively. The performance of their model was assessed using the harmonic mean score, a measure appropriate for multi-label classification tasks. The HRLCE model, when combined with BERT, achieved a harmonic mean score of 0.779, securing the 5th rank out of 165 teams in the SemEval 2019 Task 3 competition. This result highlighted the efficacy of hierarchical contextual modeling in improving emotion detection accuracy.

2.3 Emotional Intensity Analysis

Emotional intensity detection in texts is an emerging area in natural language processing (NLP) that goes beyond basic emotion recognition. It aims to quantify the degree of emotions expressed in textual data, providing a more nuanced understanding of emotional expressions. This review synthesizes research on emotion intensity detection, drawing insights from five key studies that contribute to this field.

Lexicon-Based Methods
Gupta and Yang (2018) [21] developed CrystalFeel, which predicts emotion intensity in tweets

using features derived from various affective lexicons. They used Twitter messages and derived features from parts-of-speech, n-grams, word embeddings, and multiple affective lexicons such as Opinion Lexicon, SentiStrength, AFFIN, NRC Emotion, Hash Emotion, and in-house EI Lexicons. They employed SVM-based classifier models to predict the intensity of emotions like fear, anger, sadness, and joy. CrystalFeel achieved Pearson correlations of 0.717 on average emotion intensity and 0.816 on sentiment intensity, indicating strong prediction performance facilitated by the inclusion of lexicon-based features.

Deep Learning and Multimodal Approaches

Firdaus et al. (2020) [22] introduced MEISD, a dataset for multimodal emotion and sentiment analysis in dialogues, incorporating textual, audio, and visual features. MEISD was collected from various TV series and contains balanced data representing multiple emotions with Glove Embeddings (300 dim) as the textual feature, and audio and visual features. Emotion labels were achieved via human annotation on a scale of 1 to 3. Baseline models for this dataset include multimodal neural networks (textCNN, bcLSTM, DialogueRNN, and DialogueRNN+Bert) that integrate text, audio, and video inputs to predict emotions like Joy, Sadness, Anger, Fear, Surprise, Disgust, Acceptance, and Neutral and, their intensities. The baseline models provided a foundation for future research, emphasizing the complexity and richness of multimodal data in improving emotion intensity detection.

Reader and Writer Perspective Analysis

Kajiwara et al. (2021) [23] developed WRIME, a dataset with subjective (writer's) and objective (reader's) annotations of emotional intensity in social media posts. WRIME consists of 17,000 SNS posts annotated with subjective and objective emotional intensities on a a four-point scale (0: no, 1: weak, 2: medium, and 3: strong). The emotions include anger, disgust, fear, joy, sadness, surprise, trust, and anticipation. Models for estimating emotional intensity included BoW+LogReg, fastText+SVM, and BERT models to capture the differences between subjective and objective annotations. The study highlighted the difficulty in estimating the writer's subjective emotions, particularly for anger and trust, due to a large gap between subjective and objective emotional intensities.

News Headlines Emotion Analysis

Bostan et al. (2020) [24] presented a dataset, GoodNewsEveryone, of news headlines annotated with emotions, semantic roles, and reader perceptions. The dataset comprises 5,000 English news headlines annotated with emotional content, semantic roles (emotion experiencers, cues, causes, and targets), and reader perceptions. They proposed a multiphase annotation procedure and developed baseline models (biLSTM-CRF with ELMo embeddings as input) for predicting semantic role structures and emotion intensities. Their set of 15 emotion categories is an extended set over Plutchik's emotion classes and comprises anger, annoyance, disgust, fear, guilt, joy, love, pessimism, negative surprise, optimism, positive surprise, pride, sadness, shame, and trust. Such a diverse set of emotion labels is meant to provide a more fine-grained analysis. The baseline models provided initial insights, though the task's complexity indicated the need for advanced models to accurately predict the nuanced emotional and semantic structures in news headlines.

Emotion Intensities in Social Media

Mohammad and Bravo-Marquez (2017) [25] focused on emotion intensities in tweets, providing foundational research for the field. They used a large corpus of tweets annotated for emotion intensity using Best-Worst Scaling. The features used include: Word N-grams (WN), Character N-grams (CN), Word Embeddings (WE) (Word2Vec, window size: 5; number of dimensions: 400), Affect Lexicons (L). Their approach involved an L2-regularized L2-loss SVM regression model to predict the intensity of emotions such as anger, fear, joy, and sadness with real-valued scores between 0 (lowest degree) and 1 (highest degree). Their work laid the groundwork for understanding how different emotions are expressed with varying intensities in short, informal texts like tweets. This study also contributed to the development of benchmarks for evaluating emotion intensity detection systems.

Chapter 3

Data Collection, Cleaning/Preprocessing, and Statistics

3.1 Data Collection

3.1.1 Source of Data: Reddit

Reddit, a popular social media platform, serves as an invaluable source for collecting biomedical literature due to its wide array of communities, known as subreddits, dedicated to various topics, including mental health. Reddit users frequently share their personal experiences, thoughts, and feelings in these subreddits, providing rich, authentic textual data. This user-generated content often includes detailed descriptions of symptoms, emotional states, and coping mechanisms, which are crucial for mental disorder classification. The platform's structure, with specific subreddits for different mental health disorders, allows for targeted data collection. For example, there are subreddits exclusively for anxiety, panic, and depression, each with unique discussions and support systems. This focused community interaction ensures the collected data is relevant and specific to the mental health conditions being studied.

3.1.2 Data Collection Tool: Apify

To gather data from Reddit, we used Apify, a versatile web scraping platform that offers pre-built scrapers for various social media platforms. Apify's Reddit Scraper is particularly useful as it can extract up to 1000 rows of data from specified subreddits. Apify is known for its efficiency and ease of use, providing a user-friendly interface to configure scraping tasks. The platform handles the intricacies of web scraping, such as managing requests, handling pagination, and ensuring compliance with website policies, which is crucial for obtaining large datasets without violating terms of service.

3.1.3 Data Collection Process

We focused on scraping unique posts from several popular subreddits dedicated to anxiety, panic, and depression. Only the original posts were collected, excluding comments and replies. This decision was made to ensure the primary data consisted of comprehensive, self-contained descriptions of mental health experiences, rather than fragmented or conversational snippets. The subreddits used for data collection, along with their descriptions from Reddit, are shown in Table 3.1.

Note - Subreddits like **r/mentalhealth** and **r/anxietydepression** were deliberately excluded due to their broad or mixed focus, which could introduce ambiguity into the dataset. For example, "mental health" is an umbrella term encompassing various conditions, and "r/anxiety depression" likely contains mixed posts about both anxiety and depression, complicating the labeling process.

3.1.4 Dataset Summary

After collecting data from the specified subreddits, we compiled a dataset comprising 7,866 rows. Each row represents a distinct post detailing the personal experiences and emotional states of individuals dealing with anxiety, panic, or depression. This dataset serves as the foundation for subsequent feature extraction and analysis, providing a rich textual corpus for psychometric and contextual embedding analysis.

Table 3.1: Descriptions of Subreddits Used for Data Collection

Category	**Subreddit**	**Description**
Anxiety	**r/Anxiety**	Discussion and support for individuals with anxiety disorders and their loved ones.
	r/Anxietyhelp	Scientific articles, YouTube videos, blog posts, and other resources focused on anxiety management and healing. Users are encouraged to seek professional medical assistance if experiencing a crisis.
	r/adhd_anxiety	Support for anxiety and social anxiety problems related to ADHD.
	r/socialanxiety	For distress in social situations, leading to impaired daily functioning. Symptoms include blushing, excessive sweating, trembling, palpitations, and nausea, among others.
Panic	**r/PanicAttack**	A supportive community for individuals experiencing panic attacks or panic disorders.
	r/panicdisorder	A safe space for advice, ranting, or finding support for panic sufferers.
Depression	**r/depression**	Peer support for anyone struggling with depressive disorders.
	r/depression_help	A platform for support, advice, inspiration, and motivation for individuals facing depression.

3.2 Data Cleaning/Preprocessing and Undersampling

Data cleaning is crucial in natural language processing (NLP) and data science to ensure data quality and reliability. Clean data leads to more accurate models and meaningful insights. In NLP, clean text data is essential as models are highly sensitive to noise such as duplicates, irrelevant information, and inconsistent formats. Overall, clean data helps in reducing errors, improving model performance, and ensuring reproducibility of results.

Removing Duplicates and Checking for Missing Values
The first step in our data cleaning process involved checking for duplicate rows where all columns contained identical data. We identified and removed 2 such duplicated rows. Removing duplicate rows is important as they can skew the analysis, leading to biased results and incorrect conclusions. Duplicate data can also affect the performance of machine learning models by creating redundancy and reducing the variability that the model learns from. Additionally, We conducted a thorough check to see if the text column had any `.na()` or `.isnull()` values or empty strings. Fortunately, no such rows were found. Ensuring there are no missing values is crucial because missing data can lead to errors in analysis and model training, resulting in inaccurate predictions and insights.

Filtering Texts with Less than 10 Words
We calculated the number of words in each text and removed rows where the content had fewer than 10 words, resulting in the removal of 353 rows. Texts with very few words often lack the necessary context and information for meaningful analysis and accurate classification. Such short texts can lead to poor performance of models as they provide limited linguistic cues and features for the model to learn from.

Filtering Texts with More than 1024 Words
We also removed rows where the text content had more than or equal to 1024 words, eliminating 41 rows. This was done because many transformer-based models, which we plan to use for feature extraction, have an input limit of 1024 tokens. Texts longer than this limit would need to be truncated, potentially leading to a loss of important contextual information. Ensuring that all texts fall within the acceptable length helps in maintaining the integrity of the data and allows the models to process the entire context without truncation.

Labeling the Data
The data was labeled with three categories: **"anxiety," "panic," and "depression"** according to the subreddits from which they were sourced. Proper labeling is essential for supervised learning tasks as it allows the model to learn the distinctions between different classes and make accurate predictions.

Undersampling for Balance
To achieve a balanced dataset, which is effective in the long run, we performed undersampling. A balanced dataset prevents models from being biased towards the majority class. In an imbalanced dataset, models tend to predict the majority class more frequently, leading to poor performance on the minority class. Balancing the dataset ensures that the model treats all classes equally, leading to better generalization and more robust performance.

Undersampling involves balancing uneven datasets by retaining all data from the minority class and reducing the size of the majority class. In our case, "depression" was the minority class with 1,815 rows. We undersampled the "anxiety" and "panic" classes to match this number, ensuring that data from the subreddits within each category was equally represented. After undersampling, we had 1,812 rows of unique data for "anxiety" and 1,814 for "panic". This process helped in creating a balanced dataset of 5,441 rows, crucial for training unbiased and effective machine learning models.

Figure 3.2.1 shows a preview of the final undersampled dataset.

	communityName	title	body	url	Word Count	label
0	r/Anxiety	I feel absolutely judged in every walk of life...	Literally everything I do in life I feel like ...	https://www.reddit.com/r/Anxiety/comments/1ckv...	326	anxiety
1	r/Anxiety	Somehow I ended up being a “fourth wheel”. I m...	So, it's been an awful night. \n\nThere's this...	https://www.reddit.com/r/Anxiety/comments/1cnt...	445	anxiety
2	r/Anxiety	Any advice for hyper vigilance?	Long story short I've got a lot of mental issu...	https://www.reddit.com/r/Anxiety/comments/1ckj...	151	anxiety
3	r/Anxiety	Anyone else lose their appetite when anxious?	Whenever I have stress of any kind, I feel ful...	https://www.reddit.com/r/Anxiety/comments/1cnc...	74	anxiety
4	r/Anxiety	I voted for the wrong person!	Hi everyone! So Saturday morning, I decided to...	https://www.reddit.com/r/Anxiety/comments/1clk...	72	anxiety
...	...	...	...	...	...	...
5436	r/depression_help	I'm really at wits end.	I've had family things going on for years and ...	https://www.reddit.com/r/depression_help/comme...	173	depression
5437	r/depression_help	How do I deal with negativity, apathy, and ove...	I have been trying really hard to not care wha...	https://www.reddit.com/r/depression_help/comme...	185	depression
5438	r/depression_help	How to come out of comfort zone	It's been 3 years I'm in my comfort zone, I'm ...	https://www.reddit.com/r/depression_help/comme...	77	depression
5439	r/depression_help	girlfriend wants to break up because she's dep...	for context, I (F20) and my girlfriend (F20) u...	https://www.reddit.com/r/depression_help/comme...	462	depression
5440	r/depression_help	I'm starting being afraid of myself	Everything became messy and confused in just s...	https://www.reddit.com/r/depression_help/comme...	598	depression

5441 rows × 6 columns

Figure 3.2.1: Undersampled Dataset

3.3 Dataset Characteristics and Word Cloud Analysis

3.3.1 Initial Features

Initially, we have four features/columns in our dataset:

- **communityName:** This column contains the name of the subreddit from which the post was collected. Subreddits are specific communities on Reddit, each dedicated to a particular topic or theme. In our case, the subreddits are focused on mental health issues such as anxiety, panic, and depression.
- **title:** Every post on Reddit has a title that gives a brief overview of the content. This title is often a summary or a highlight of the main post. The title is useful for understanding the context of the post at a glance.
- **body:** This is the main content of the Reddit post. It contains detailed textual data where users share their thoughts, experiences, and emotions. This is the most crucial part of the dataset as it holds the information needed for our analysis.
- **url:** This is the link to the original Reddit post. Having the URL is helpful for referencing and verifying the source of the data. It allows us to revisit the original post if further information or context is needed.

3.3.2 Dataset Statistics

After collecting the data, we performed undersampling to balance the dataset across the different labels. The label and subreddit distribution before and after the undersampling are provided in Tables 3.2 and 3.3 below.

Table 3.2: Before Undersampling

Label	**Count**
Anxiety	3743
Panic	1912
Depression	1815

Subreddit Name	**Count**
r/Anxiety	967
r/Anxietyhelp	911
r/adhd_anxiety	933
r/socialanxiety	932
r/PanicAttack	945
r/panicdisorder	967
r/depression	938
r/depression_help	877

Table 3.3: After Undersampling

Label	**Count**
Anxiety	1812
Panic	1814
Depression	1815

Subreddit Name	**Count**
r/Anxiety	453
r/Anxietyhelp	453
r/adhd_anxiety	453
r/socialanxiety	453
r/PanicAttack	907
r/panicdisorder	907
r/depression	938
r/depression_help	877

3.3.3 Word Cloud Analysis

A Word Cloud is a visual representation of text data. Words from the text are displayed in different sizes, where the size of each word indicates its frequency or importance within the text. Commonly occurring words are shown in larger fonts, while less frequent words appear in smaller fonts. They provide an immediate visual cue to the most significant words in a dataset, helping to quickly identify prominent themes and topics and serving as a summary tool for large text corpora.
To gain insights into the content of our dataset, we created word clouds for the entire dataset as well as for each of the three labels: anxiety, panic, and depression.
Note - We removed the stopwords (common words like "the," "is," "and") from the text before creating the word clouds.

Entire Dataset Word Cloud

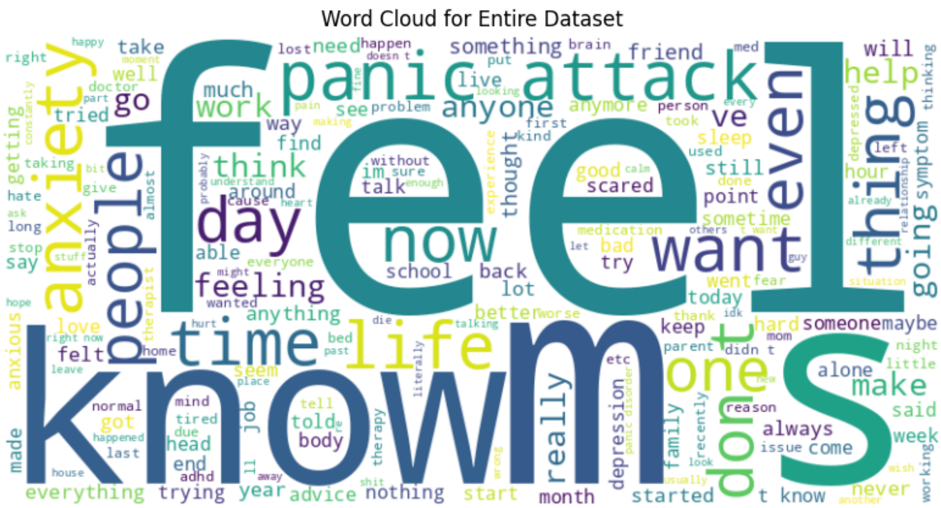


Figure 3.3.1: Entire Dataset Word Cloud

Figure 3.3.1: The word cloud for the entire dataset showcases the most common words across all posts, without distinguishing between specific mental health issues. Prominent words include **"feel," "anxiety," "panic," "depression," "time," "help," "life," "people," and "know."** These words suggest a broad range of emotions and situations that users are discussing. The frequent appearance of terms like "anxiety," "panic," and "depression" highlights the focus on mental health issues. Words like "feel" and "help" indicate a strong presence of discussions around personal emotions and the need for support.

Anxiety Word Cloud

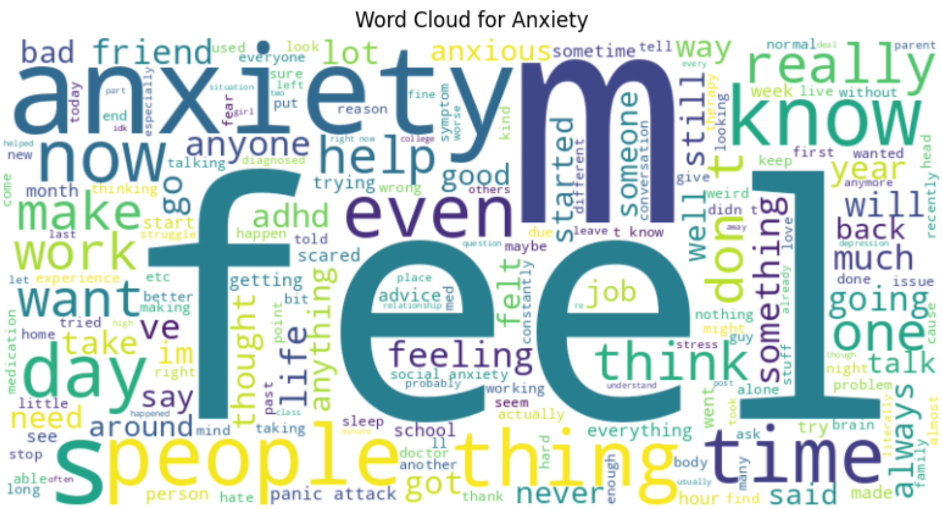


Figure 3.3.2: Anxiety Word Cloud

Figure 3.3.2: For posts labeled with anxiety, prominent words include **"anxiety," "feel," "time," "people," "help," "work," "life," and "stress."** The central term "anxiety" confirms the focus of these discussions. Words like "feel," "time," and "people" reflect the emotional and social dimensions of anxiety, highlighting how it affects daily life and interactions. "Work" and "stress" suggest common triggers or related factors, indicating that anxiety is often discussed in the context of workplace stress and overall life management. "Help" again signifies the need for support and coping mechanisms.

Panic Word Cloud

Figure 3.3.3: Panic Word Cloud

Figure 3.3.3: The word cloud for posts labeled with panic reveals words such as **"panic," "attack," "feel," "heart," "time," "help," "anxiety," and "breathing."** These words are closely related to the symptoms and experiences of panic attacks. "Panic" and "attack" are expectedly prominent, emphasizing the focus of these posts. Words like "heart," "breathing," and "feel" suggest the physical symptoms associated with panic attacks, such as palpitations and shortness of breath. "Help" and "anxiety" indicate the frequent overlap with anxiety and the seeking of assistance or support during these episodes.

Depression Word Cloud

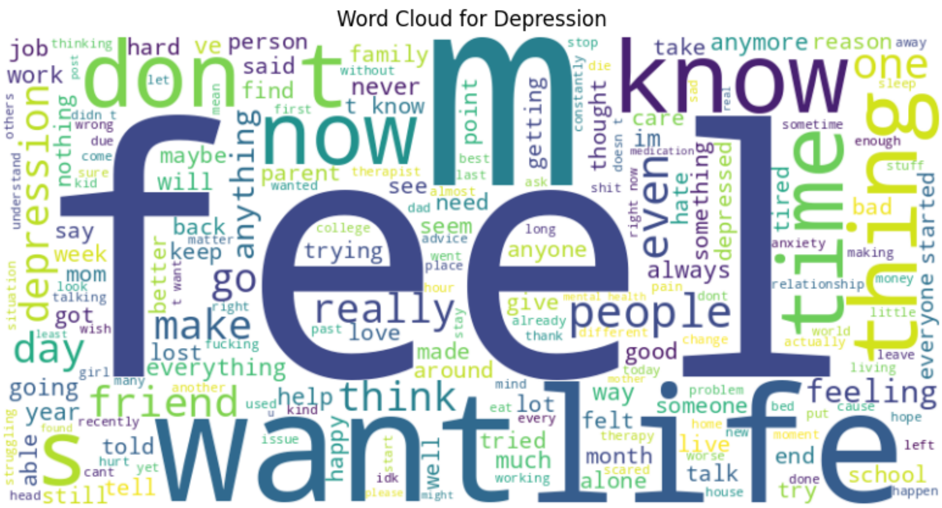


Figure 3.3.4: Depression Word Cloud

Figure 3.3.4: The word cloud for depression posts features words such as **"depression," "feel," "life," "time," "people," "help," "sad," and "think."** "Depression" is prominently central, indicating the primary focus. Words like "feel," "life," and "time" point to the pervasive impact of depression on an individual's emotions and overall life experience. "Sad" reflects the emotional

state commonly associated with depression, while "think" may indicate introspection or negative thought patterns. "Help" signifies the recurring theme of seeking assistance, and "people" suggests the social dimension, potentially highlighting feelings of isolation or the impact of relationships on mental health.

Chapter 4

Feature Extraction

4.1 Contextual Embeddings

4.1.1 Definition and Importance of Contextual Embeddings in NLP

In natural language processing (NLP), contextual embeddings refer to vector representations of words, sentences, or paragraphs that capture semantic meaning within the context in which they appear. Unlike traditional word embeddings, such as Word2Vec or GloVe, which assign a single vector to each word regardless of context, contextual embeddings dynamically adjust these representations based on surrounding words. This context-awareness enables a more nuanced understanding and processing of textual data.

The transformation of textual data into numerical arrays, or embeddings, is fundamental for feeding into machine learning models. Each element of the resulting array represents various semantic and syntactic features of the text, facilitating complex tasks such as classification, sentiment analysis, and named entity recognition.

4.1.2 Application of Contextual Embeddings

For our project, we have utilized SentenceTransformers (a.k.a. SBERT) [26], a powerful tool for generating sentence embeddings. It is the go-to Python module for accessing, using, and training state-of-the-art text and image embedding models. It can be used to compute embeddings using Sentence Transformer models or to calculate similarity scores using Cross-Encoder models. This unlocks a wide range of applications, including semantic search, semantic textual similarity, and paraphrase mining. SentenceTransformers is built on top of popular transformer models and is available through HuggingFace, a prominent platform offering a wide array of pre-trained NLP models.

Characteristics of Sentence Transformer (a.k.a bi-encoder) models:

- Calculates a fixed-size vector representation (embedding) given texts or images.
- Embedding calculation is often efficient, and embedding similarity calculation is very fast.
- Applicable for a wide range of tasks, such as semantic textual similarity, semantic search, clustering, classification, paraphrase mining, and more.

A wide selection of over 5,000 pre-trained Sentence Transformers models are available for immediate use on HuggingFace. Once the library is installed, any of the models can be accessed

```
[ ] !pip install sentence-transformers
    from sentence_transformers import SentenceTransformer
    model = SentenceTransformer('bert-base-nli-mean-tokens')

[ ] sentences = ['This framework generates embeddings for each input sentence',
        'Sentences are passed as a list of string.',
        'The quick brown fox jumps over the lazy dog.']

[ ] sentence_embeddings = model.encode(sentences)

[ ] sentence_embeddings

    array([[-0.10409462,  0.52747655,  1.1797726 , ..., -0.4338911 ,
            -0.69452405,  0.5386931 ],
           [-0.13118464, -0.17390314,  1.1052175 , ...,  0.02624492,
            -0.00269875,  0.91611093],
           [-0.74899244,  0.71891886, -1.0394562 , ...,  0.15582585,
             1.020252  ,  0.09790432]], dtype=float32)

[ ] len(sentence_embeddings)

    3

[ ] len(sentence_embeddings[0])

    768
```

Figure 4.1.1: SentenceTransformers Code Implementation

to retrieve the embeddings of a text, as shown in Figure 4.1.1.

After testing several models, **all-distilroberta-v1** yielded the best results in terms of classification performance, making it the chosen model for subsequent experiments (discussed in detail in Chapter 6, Section 6.1.1 (Sentence Embeddings Results)). It maps sentences and paragraphs to a 768 dimensional dense vector space which captures the semantic information. It used the pre-trained distilroberta-base model [27], and using the concatenation from multiple datasets, the model was fine-tuned on a 1B sentence pairs dataset. The model was trained on a TPU v3-8, during 920k steps using a batch size of 512 (64 per TPU core). The learning rate warm-up of 500 and the AdamW optimizer with a 2e-5 learning rate were used.

In conclusion, contextual embeddings play a vital role in transforming textual data into a format suitable for machine learning models. Through rigorous evaluation, all-distilroberta-v1 was selected for its optimal performance on our specific dataset, underscoring the importance of model selection in NLP tasks.

4.2 Linguistic Inquiry and Word Count (LIWC) - 22 Features

4.2.1 Introduction of LIWC-22

People reveal themselves through the words they use, and analyzing this language can provide deep insights into their thoughts, feelings, personality, and social connections. Words used in everyday life reflect psychological states such as beliefs, emotions, thinking habits, experiences, relationships, and personality traits. From Freud's "slips of the tongue" to modern computer-based text analysis, extensive research across social sciences has demonstrated the significant psychological value of words.

Linguistic Inquiry and Word Count (LIWC) is text analysis software that quantifies various language dimensions, offering profound insights into the corpus. The latest version, LIWC-22 [28], categorizes words into psychologically meaningful categories and outputs various linguistic and psychological features. Widely used in computational linguistics, psychology, and other fields, LIWC-22 (Pennebaker et al., 2022) has updated its dictionary and software options to align with new text analysis directions. The program is designed for quick and efficient analysis of individual or multiple language files while maintaining transparency and flexibility, allowing users to explore word use in multiple ways.

4.2.2 Importance in NLP and Biomedical Literature

From the perspective of Natural Language Processing (NLP), LIWC-22 is significant because it provides insights into the psychological and emotional states conveyed through language. It allows researchers to quantify abstract psychological constructs such as emotional tone, social concerns, and cognitive processes, which are otherwise difficult to measure. These features can enrich text data with additional layers of meaning and improve the performance of various NLP applications, including sentiment analysis, emotion detection, and mental health assessments.
LIWC has been extensively used in biomedical literature to study various aspects of mental health and psychological well-being. It has been used to extract lexico-syntactic features as a baseline measure to detect anxiety in social media posts [29]. During COVID-19, people extensively wrote about their concerns and expressed themselves through texts on social media platforms such as Reddit and Twitter. Viviani et al. (2021) [30] used a sub-category of LIWC features to assess vulnerability to psychological distress during the COVID-19 pandemic through the analysis of microblogging content. Another study used LIWC to extract signs of depression and suicidal ideation from college students [31].

4.2.3 Types of LIWC outputs and their Interpretation

LIWC calculates word counts and categorizes them into several psychological groups. It identifies significant words in texts and sorts them into two main categories: traditional LIWC dimensions and Summary Measures/Variables.

Most output variables fall under traditional LIWC dimensions, which indicate the percentage of total words in a text for each dimension. For example, if a blog analysis shows a Positive Emotions (emo_pos) score of 6.30, it means 6.30 percent of the words in the blog are positive emotion words.

This category includes 112 dimensions, such as I-Words (I, Me, My, Myself), You-Words (You, Your, Yourself), Social Words, Positive Emotions, Negative Emotions, and Cognitive Processes. Other abbreviations include tone_pos, tone_neg, emotion, emo_pos, emo_neg, emo_anx, emo_anger, emo_sad, swear, socbehav, polite, conflict, moral, family, friend, female, male, Culture, politic, ethnicity, tech, Lifestyle, leisure, home, work, money, relig, Physical, health, illness, wellness, mental, substances, sexual, food, death, and more.

A few LIWC variables are calculated differently: word count (WC), words per sentence (WPS), and summary measures. WC is the raw number of words in a file. WPS is the average number of words per sentence in the file.

The second major category is Summary Measures, which includes four domains: Analytic (Analytical Thinking), Clout, Authenticity, and Emotional Tone. The analytic domain relates to formal thinking; clout refers to authoritativeness, confidence, and leadership capabilities; authenticity indicates the truthfulness of the writing; emotional tone reflects the emotional state of the participant based on the text. Each summary measure is an algorithm derived from various LIWC variables based on previous empirical research. The numbers are standardized scores converted to percentiles (ranging from 1 to 99) based on the area under a normal curve. Detailed descriptions of the summary measures are as follows -

- **Analytic:** The analytical thinking variable measures the extent to which individuals use words indicative of formal, logical, and hierarchical thinking patterns. Individuals low in analytical thinking tend to use more intuitive and personal language, while those high in this dimension use language that is often rewarded in academic settings and associated with higher grades and reasoning skills. Language low in analytical thinking is perceived as more friendly and personable, whereas high analytical language is seen as more structured and formal.

- **Clout:** Clout measures the relative social status, confidence, or leadership displayed through writing or speech. It differs from the concept of "Power" (as defined by the LIWC-22 "power" variable), which refers more to an individual's attention to or awareness of social status. Clout specifically reflects the confidence or dominance shown in language use.

- **Authenticity:** Authenticity captures how genuinely individuals reveal themselves through their language. High authenticity is characterized by spontaneous, unfiltered speech, often seen in casual conversations between close friends or candid political speeches. Low authenticity is typical in prepared texts, such as pre-written speeches, or when individuals are being socially cautious and self-regulating their language.

- **Emotional Tone:** The emotional tone variable in LIWC-22 combines both positive and negative tone dimensions into a single summary metric. A higher score indicates a more positive tone, while scores below 50 suggest a more negative emotional tone.

Figure 4.2.1 shows the first few LIWC features and some words associated with each dimension.

Category	Abbrev.	Description/Most frequently used exemplars
Summary Variables		
Word count	WC	Total word count
Analytical thinking	Analytic	Metric of logical, formal thinking
Clout	Clout	Language of leadership, status
Authentic	Authentic	Perceived honesty, genuineness
Emotional tone	Tone	Degree or positive (negative) tone
Words per sentence	WPS	Average words per sentence
Big words	BigWords	Percent words 7 letters or longer
Dictionary words	Dic	Percent words captured by LIWC
Linguistic Dimensions	Linguistic	
Total function words	function	the, to, and, I
Total pronouns	pronoun	I, you, that, it
Personal pronouns	ppron	I, you, my, me
1st person singular	i	I, me, my, myself
1st person plural	we	we, our, us, lets
2nd person	you	you, your, u, yourself
3rd person singular	shehe	he, she, her, his
3rd person plural	they	they, their, them, themsel*
Impersonal pronouns	ipron	that, it, this, what
Determiners	det	the, at, that, my
Articles	article	a, an, the, alot
Numbers	number	one, two, first, once
Prepositions	prep	to, of, in, for
Auxiliary verbs	auxverb	is, was, be, have
Adverbs	adverb	so, just, about, there
Conjunctions	conj	and, but, so, as
Negations	negate	not, no, never, nothing
Common verbs	verb	is, was, be, have
Common adjectives	adj	more, very, other, new
Quantities	quantity	all, one, more, some
Psychological Processes		
Drives	Drives	we, our, work, us
Affiliation	affiliation	we, our, us, help
Achievement	achieve	work, better, best, working
Power	power	own, order, allow, power

Figure 4.2.1: LIWC features and examples of associated words

4.2.4 LIWC Analysis

Personal Pronouns Usage

The analysis revealed that the mean value of the usage of **"i" words (I, Me, My, Myself)** in texts was highest for **depression (12.45)**, followed by **anxiety (10.72)**, and **panic (10.18)**. Similarly, the mean value of the usage of **"we" words (we, our, us, lets)** was also highest for **depression (15.08)**, compared to **anxiety (12.93)** and **panic (11.93)**. This suggests that individuals experiencing depression tend to use more personal pronouns in their language while expressing their thoughts. The increased use of personal pronouns might indicate a greater focus on self and personal experiences, which is often characteristic of depressive states where individuals are more introspective and self-reflective.

Negative Emotion and Tone
The mean value of the usage of **"emo_neg" words (bad, hate, hurt, tired)** was highest for **panic (4.98)**, followed by **anxiety (3.05)**, and **depression (2.38)**. Similarly, the usage of **"emotion" words** was also highest for **panic (5.66)**, compared to **anxiety (3.79)** and **depression (3.36)**. Furthermore, the usage of **"tone_neg" words (bad, wrong, too much, hate)** was highest for **panic (6.31)**, followed by **anxiety (4.30)**, and **depression (4.02)**. These findings indicate that individuals experiencing panic tend to use more emotional language, especially with a negative tone and negative emotions. This might be because panic attacks are often associated with intense and immediate emotional responses, leading individuals to express more negative emotions and tones in their language.

Sadness Expression
The mean value of the usage of **"emo_sad" words (:(, sad, disappoint, cry)** was highest for **depression (0.85)**, compared to **anxiety (0.23)** and **panic (0.19)**. This indicates that individuals with depression tend to use more words expressing sadness and unhappiness in their language. This is consistent with the symptoms of depression, which often include persistent feelings of sadness and hopelessness.

Death and Family
The mean value of the usage of **"death" words (death, dead, die, kill)** was highest for **depression (0.38)**, followed by **panic (0.18)**, and **anxiety (0.11)**. Additionally, the usage of **"family" words (parent, mother, father, baby)** was also highest for **depression (0.50)**, compared to **anxiety (0.24)** and **panic (0.16)**. This suggests that individuals with depression tend to use more words related to death and family. The focus on death might be indicative of suicidal ideation or existential thoughts, while the emphasis on family could reflect concerns or emotional ties to family members, which are often significant in the context of depression.

Health-Related Terms
The mean value of the usage of **"mental" words (mental health, depressed, suicide, trauma)** was highest for **panic (2.70)**, compared to **depression (0.97)** and **anxiety (0.89)**. Similarly, the usage of **"illness" words (hospital, cancer, sick, pain)** was highest for **panic (0.81)**, followed by **anxiety (0.42)** and **depression (0.26)**. The usage of **"health" words (medic, patients, physician, health)** was also highest for **panic (4.94)**, compared to **anxiety (2.43)** and **depression (2.02)**. Finally, the usage of **"physical" words (medic, food, patients, eye)** was highest for **panic (7.68)**, followed by **anxiety (4.23)** and **depression (3.52)**. These findings indicate that individuals experiencing panic tend to use more words related to mental and physical health, illnesses, and general health. This might be because panic attacks often involve acute physical symptoms and concerns about health, leading individuals to express these topics more frequently in their language.

Desire and Wants
The mean value of the usage of **"want" words (want, hope, wanted, wish)** was highest for **depression (0.91)**, followed by **anxiety (0.49)** and **panic (0.33)**. This suggests that individuals with depression tend to express more desires and wishes in their language. This might reflect a longing for change or improvement in their situation, which is common among those experiencing depressive symptoms.

4.3 Emotion Detection

4.3.1 Definition and Importance of Emotion Detection in Texts

Emotion detection in texts involves identifying and classifying the emotional tone conveyed by the text. This process can reveal insights into the psychological state and sentiments of the author, which can be particularly useful in various applications, such as Mental Health Monitoring, Customer Feedback Analysis, Social Media Monitoring, and Human-Computer Interaction.

4.3.2 Choice of Model: EmoRoBERTa

For the task of emotion detection in biomedical texts, we used the EmoRoBERTa model available on HuggingFace. EmoRoBERTa is a specialized model fine-tuned for emotion detection tasks, and it is built upon the RoBERTa (A Robustly Optimized BERT Pretraining Approach) architecture. Several factors influenced this choice:

- **Wide Range of Emotions:** EmoRoBERTa can detect 28 distinct emotions, providing a comprehensive analysis of emotional states.
- **Popularity and Community Validation:** The model has been extensively downloaded and liked on the HuggingFace platform, indicating its reliability and effectiveness.
- **Community and Usage:** The wide adoption of EmoRoBERTa by the HuggingFace community further validates its robustness and applicability in various text analysis tasks.

4.3.3 Training Dataset: GoEmotions

EmoRoBERTa is trained on the GoEmotions dataset [32], which is one of the most comprehensive emotion datasets available. Key aspects of the dataset include:

- **Source:** The GoEmotions dataset is composed of 58,000 Reddit comments, providing a rich and diverse set of text samples.
- **Labels:** The dataset is annotated with 28 emotion labels, making it suitable for multi-label classification tasks.
- **Multi-label Nature:** Each comment can be associated with multiple emotion labels, reflecting the complexity of human emotions. The EmoRoBERTa model is, therefore, designed as a multi-label classification model, outputting 28 probability scores, each representing the likelihood of a specific emotion being present in the text. The softmax function ensures that the sum of these probabilities is 1.

4.3.4 Detected Emotions

EmoRoBERTa is capable of detecting the following 28 emotions:
admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise, neutral

An example of two different types of texts and their emotions extraction is shown in Figure 4.3.1. As evident from Figure 4.3.1 (a), for a positive text, emotions such as **amusement, excitement, and joy** show high probabilities of **0.44, 0.15, and 0.10,** respectively. Similarly, from Figure 4.3.1 (b), for a negative text, emotions such as **grief (0.32)** and **sadness (0.23)** are more prevalent.

```
Text: The movie was so hilarious, it had me laughing uncontrollably.
Extracted Emotions:
admiration: 0.01294624526053667
amusement: 0.4414743185043335
anger: 0.0031005926430225372
annoyance: 0.004050253424793482
approval: 0.02912912145256996
caring: 0.029954388737678528
confusion: 0.005281085148453712 5
curiosity: 0.02762742154300213
desire: 0.02023329958319664
disappointment: 0.0016962970839813352
disappproval: 0.00553554343059659
disgust: 0.0014652491081506014
embarrassment: 0.0035885379184037447
excitement: 0.15490929782390594
fear: 0.0031673626508563757
gratitude: 0.008031763136386871
grief: 0.0034536407329142094
joy: 0.103614442050457
love: 0.007730493322014809
nervousness: 0.007319073658436537
optimism: 0.011901124380528927
pride: 0.013524687848985195
realization: 0.019867902621626854
relief: 0.019469087943434715
remorse: 0.011299205012619495
sadness: 0.0010723805753514 17
surprise: 0.03680050000548363
neutral: 0.011756670661270618
```

(a) Emotion Extraction of a positive text

```
Text: This news has left me utterly devastated.
Extracted Emotions:
admiration: 0.0016848123632371426
amusement: 0.0009099639719352126
anger: 0.008117130026221275
annoyance: 0.002892033429816365 2
approval: 0.004944671876728535
caring: 0.012212402187287807
confusion: 0.014136777259409428
curiosity: 0.005468911956995726
desire: 0.006198229268193245
disappointment: 0.09196507185697556
disapproval: 0.028957227244973183
disgust: 0.03895663097500801
embarrassment: 0.03178337961435318
excitement: 0.0020342145580798388
fear: 0.025961806997656822
gratitude: 0.001343477051705122
grief: 0.3230440616607666
joy: 0.0008202795870602131
love: 0.002255234401673078 5
nervousness: 0.006387366447597742
optimism: 0.0008344837697222829
pride: 0.0020536796655505896
realization: 0.022880850359797478
relief: 0.0010854427237063646
remorse: 0.10942316800355911
sadness: 0.2336918711662292 5
surprise: 0.016586998477578163
neutral: 0.0033697790931910276
```

(b) Emotion Extraction of a negative text

Figure 4.3.1: Emotion Extraction of two different types of texts

Code for Emotion extraction -

```
[ ] !pip install transformers
    import tensorflow as tf
    from transformers import RobertaTokenizerFast, TFRobertaForSequenceClassification

    tokenizer = RobertaTokenizerFast.from_pretrained("arpanghoshal/EmoRoBERTa")
    model = TFRobertaForSequenceClassification.from_pretrained("arpanghoshal/EmoRoBERTa")

    text_list = list(df['text'])
    emo_label_prob = []

    for item in text_list:
      encoded_input = tokenizer(item, truncation=True, padding=True, return_tensors="tf")
      output = model(encoded_input["input_ids"], attention_mask=encoded_input["attention_mask"])

      emotions = output.logits[0].functional.softmax()
      emotions_prob = tf.math.softmax(emotions).numpy()

      emotion_labels = model.config.id2label.values()
      emotion_probabilities = dict(zip(emotion_labels, emotions_prob))

      emo_label_prob.append(emotion_probabilities)

    print(emo_label_prob)
```

Figure 4.3.2: Emotions Extraction Code Implementation

4.3.5 Emotions Analysis

In this analysis, we explore the distribution of emotions. The dataset consists of textual data along with corresponding probabilities of 28 emotions. We begin by calculating the mean probabilities of each emotion across the entire dataset. Further, we delve into the distribution of emotions within each label category (anxiety, panic, depression). By visualizing this data, we gain insights into the overall emotional tone of the text corpus.

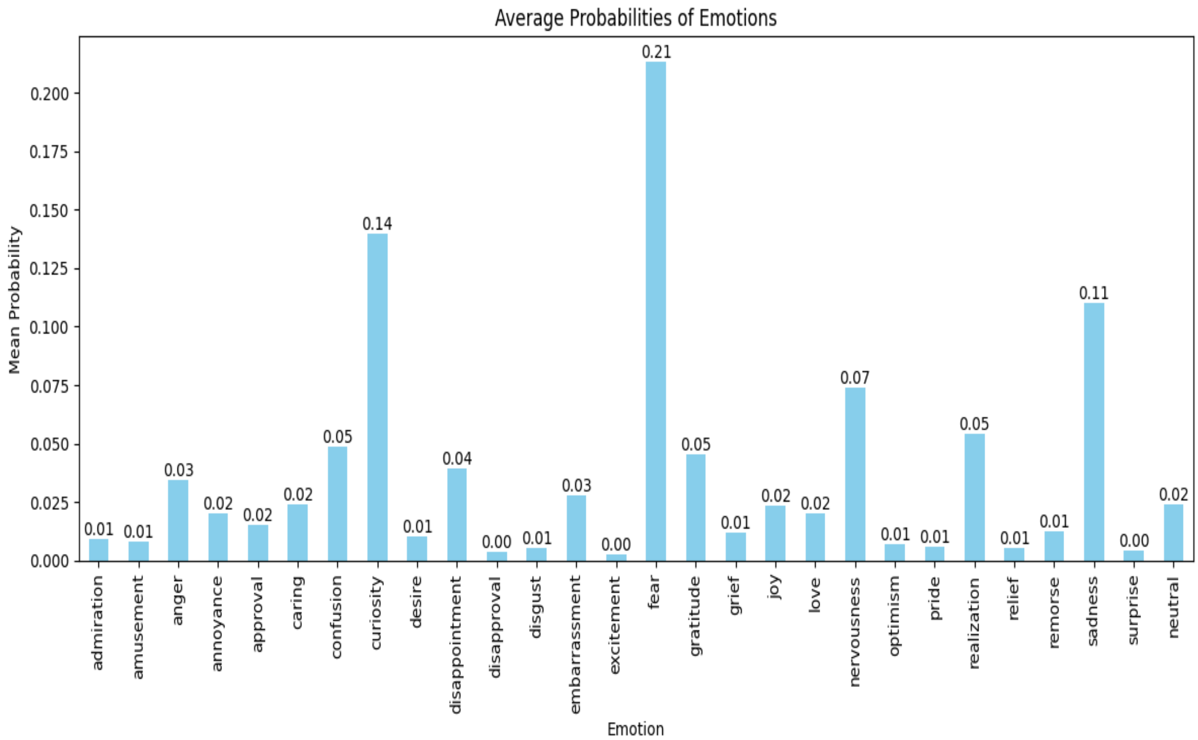


Figure 4.3.3: Average Probabilities of emotions for entire dataset

Figure 4.4.3: In the entire dataset, **fear** emerges as the most prominent emotion with a probability of **0.21,** indicating a prevailing sense of danger or concern across various contexts. **Curiosity** closely follows with a probability of **0.14,** suggesting a strong interest and desire for understanding and seeking knowledge. **Sadness (0.11)** reflects content evoking sorrow or unhappiness, while **nervousness (0.07)** contributes to an atmosphere of discomfort and unease. Additionally, **gratitude (0.05),** though less prevalent, signifies moments of positivity and appreciation amidst the prevailing fear and sadness.

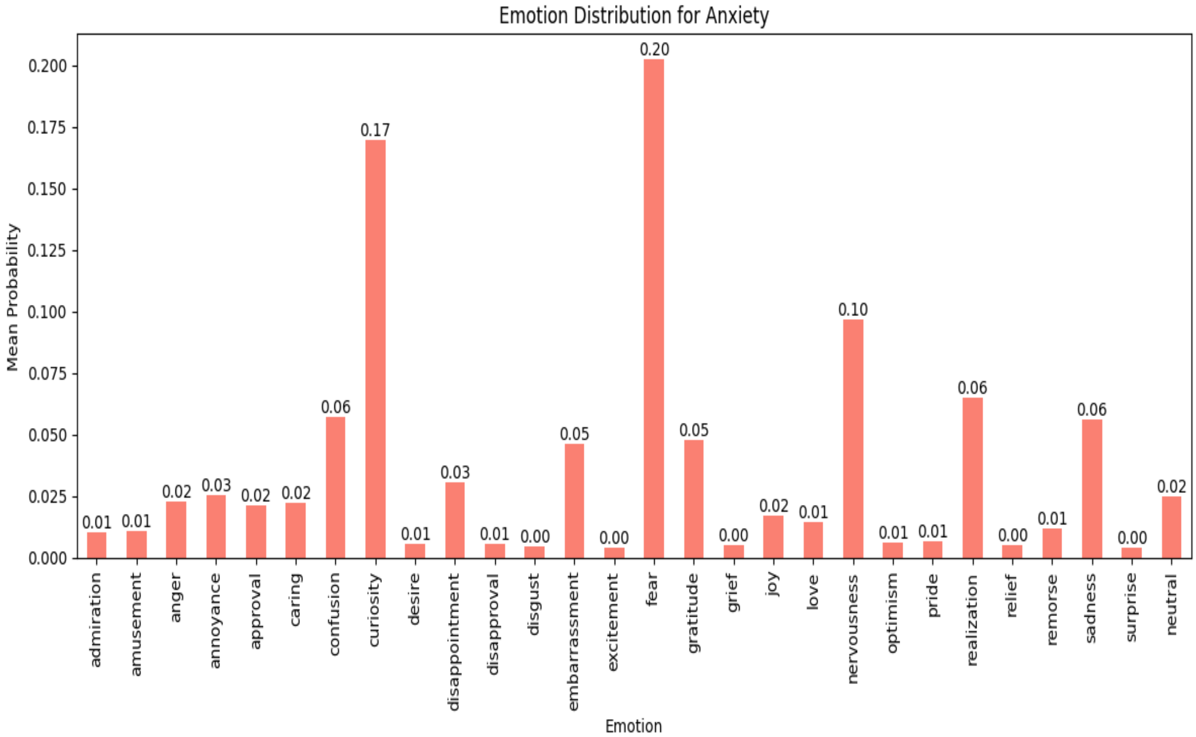


Figure 4.3.4: Average Probabilities of Emotions for Anxiety dataset

Figure 4.3.4: In the anxiety dataset, **fear** is the most prominent emotion with a probability of **0.20. Curiosity (0.17)** follows closely behind, again indicating that people are willing to understand more about mental disorders. **Nervousness (0.10)** reflects the characteristic restlessness and heightened alertness associated with anxiety, while **confusion (0.06)** underscores the uncertainty and lack of clarity often experienced, highlighting the struggle of dealing with unclear situations contributing to anxiety.

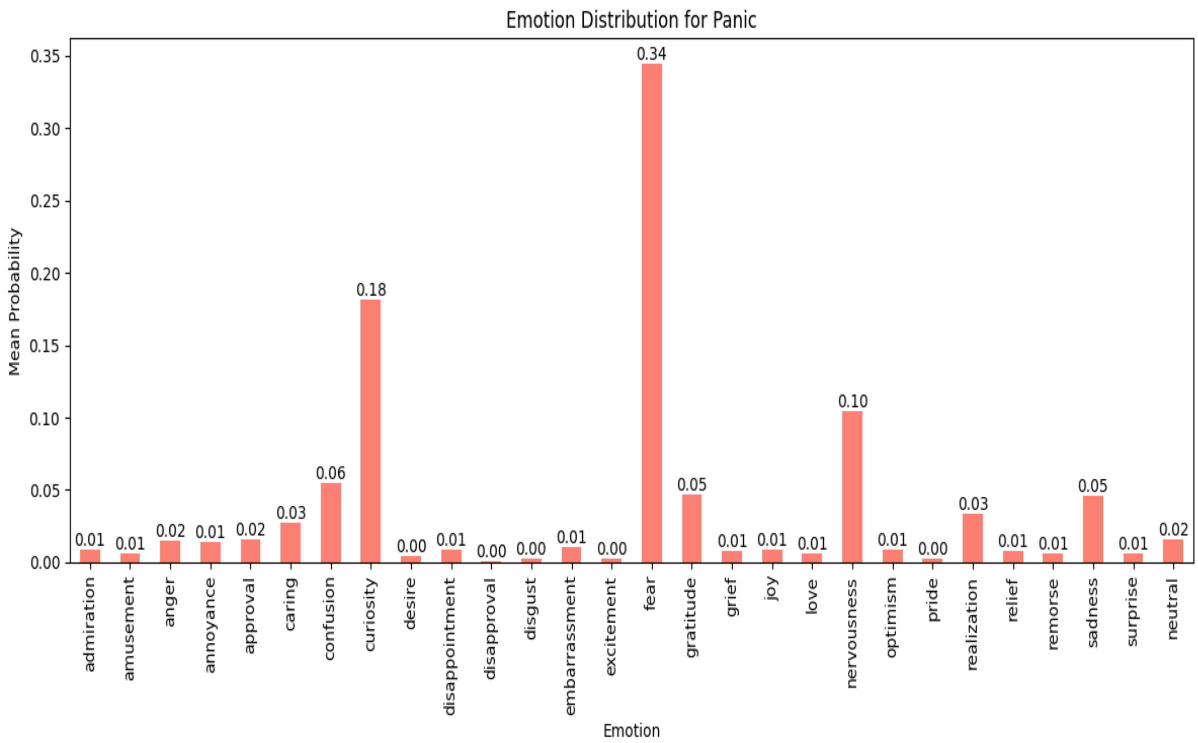


Figure 4.3.5: Average Probabilities of Emotions for Panic dataset

Figure 4.3.5: In the panic dataset, **fear (0.34)** overwhelmingly emerges as the most prominent emotion, reflecting the intense fear and sense of imminent danger characteristic of panic states. **Curiosity (0.18)** and **Nervousness (0.10)** still remain significant. While **confusion at 0.06** and **sadness at 0.05** underscore uncertainty and unhappiness, often experienced during panic attacks, **gratitude at 0.05** suggests moments of positive interactions amidst panic situations.

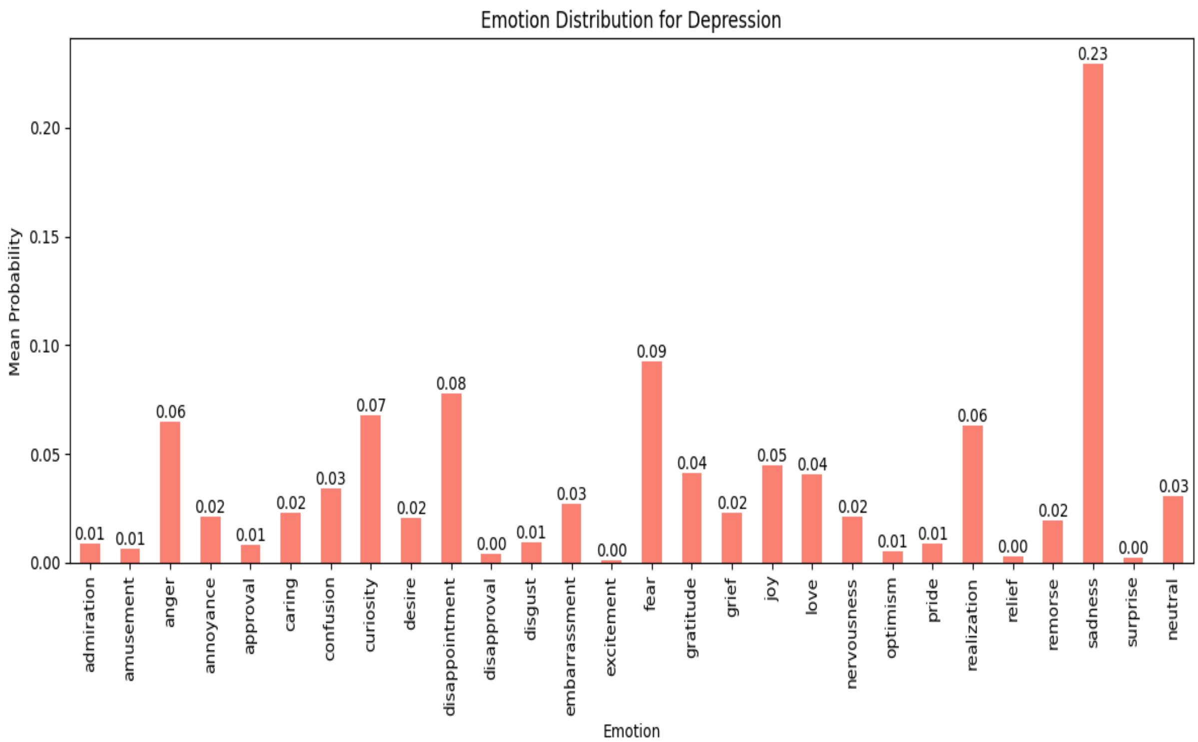


Figure 4.3.6: Average Probabilities of Emotions for Depression dataset

Figure 4.3.6: In the depression dataset, **sadness (0.23)** emerges as the most prominent emotion, reflecting pervasive feelings of sorrow and hopelessness characteristic of depression. **Fear (0.09),** though less prominent, suggests underlying concerns and worries contributing to the depressive state. **Disappointment (0.08)** follows closely behind, indicating frequent experiences of unmet expectations and frustration, which are common in depressive states. Additionally, **realization (0.06)** suggests moments of clarity or epiphany within the depressive state, where individuals might gain insights into their emotions or situations.

Conclusion
Emotion detection using models like EmoRoBERTa provides valuable insights into the emotional content of texts. By leveraging a robust model trained on a comprehensive dataset, we can achieve accurate and nuanced emotion classification, supporting various research and practical applications.

4.4 Emotion Intensity

4.4.1 Definition of Emotion Intensity

Emotion intensity refers to the strength or magnitude of an emotional experience expressed in text. Unlike simple emotion detection, which identifies the presence of an emotion, emotion intensity quantifies how strongly that emotion is conveyed. Words can convey different intensities of emotion. For instance, most people would agree that "enrage" expresses a stronger degree of anger than "annoy." However, annotating words for nuanced emotional intensity is much more challenging than simple categorical annotation. This task imposes a greater cognitive load on respondents and makes it difficult to maintain consistency both across different annotators and within the responses of the same annotator. The Effect Intensity lexicon, which assigns real-valued scores to words based on their emotional intensity, was developed using best-worst scaling.

4.4.2 NRC Emotion Intensity Lexicon (NRC-EIL)

For this project, we employed a lexicon-based approach using the NRC Emotion Intensity Lexicon (version 1) [33]. The lexicon contains 9,829 English words with real-valued intensity scores for eight basic emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, and trust), which Robert Plutchik identified as fundamental or universal. It includes common English terms as well as those more frequently used on social media platforms like Twitter. The lexicon features terms associated with these emotions to varying degrees. For a specific emotion, it even includes some terms that might not predominantly express that emotion (or might convey an opposite emotion) but tend to co-occur with terms that do. Antonymous terms often co-occur more frequently than by chance, posing challenges for automatic co-occurrence-based statistical methods used to capture word-emotion connotations.

For a given word w and emotion e, the scores range from 0 to 1.

- A score of 1 means that the w conveys the highest amount of emotion e.
- A score of 0 means that the w conveys the lowest amount of emotion e.

Example entries from the lexicon are shown in Figure 4.4.1.

Word	Anger	Word	Fear	Word	Joy	Word	Sadness
outraged	0.964	*horror*	0.923	*sohappy*	0.868	*sad*	0.844
brutality	0.959	*horrified*	0.922	*superb*	0.864	*suffering*	0.844
satanic	0.828	*hellish*	0.828	*cheered*	0.773	*guilt*	0.750
hate	0.828	*grenade*	0.828	*positivity*	0.773	*incest*	0.750
violence	0.742	*strangle*	0.750	*merrychristmas*	0.712	*accursed*	0.697
molestation	0.742	*tragedies*	0.750	*bestfeeling*	0.712	*widow*	0.697
volatility	0.687	*anguish*	0.703	*complement*	0.647	*infertility*	0.641
eradication	0.685	*grisly*	0.703	*affection*	0.647	*drown*	0.641
cheat	0.630	*cutthroat*	0.664	*exalted*	0.591	*crumbling*	0.594
agitated	0.630	*pandemic*	0.664	*woot*	0.588	*deportation*	0.594
defiant	0.578	*smuggler*	0.625	*money*	0.531	*isolated*	0.547
coup	0.578	*pestilence*	0.625	*rainbow*	0.531	*unkind*	0.547
overbearing	0.547	*convict*	0.594	*health*	0.493	*chronic*	0.500
deceive	0.547	*rot*	0.594	*liberty*	0.486	*injurious*	0.500
unleash	0.515	*turbulence*	0.562	*present*	0.441	*memorials*	0.453
bile	0.515	*grave*	0.562	*tender*	0.441	*surrender*	0.453
suspicious	0.484	*failing*	0.531	*warms*	0.391	*beggar*	0.422
oust	0.484	*stressed*	0.531	*gesture*	0.387	*difficulties*	0.421
ultimatum	0.439	*disgusting*	0.484	*healing*	0.328	*perpetrator*	0.359
deleterious	0.438	*hallucination*	0.484	*tribulation*	0.328	*hindering*	0.359

Figure 4.4.1: Example entries for four (of the eight) emotions in the NRC Emotion Intensity Lexicon. For each emotion, the table shows every 100th and 101th entry, when ordered by decreasing emotion intensity.

The number of lexicon words associated with each of the 8 emotions is shown in Figure 4.4.2.

```
[ ] lexicons_df['lex_emotion'].value_counts()

fear            1763
trust           1490
anger           1481
sadness         1294
joy             1264
disgust         1092
anticipation     862
surprise         583
Name: lex_emotion, dtype: int64
```

Figure 4.4.2: Number of lexicon words per emotion

4.4.3 Implementation

To integrate emotion intensity analysis into our project, we followed these steps:

1. **Preprocessing**: Convert all text in our dataset to lowercase. This step ensures consistency since all words in the NRC-EIL are in lowercase.

2. **Calculating Emotional Intensities**:

   - For each word in a given text, check if the word exists in the NRC-EIL.
   - If the word is not found in the lexicon, it does not contribute to the emotional intensity calculation and is ignored.
   - If the word is found, retrieve the associated emotion and their respective intensity score from the lexicon and store it in a dictionary.
   - This process is repeated for all eight emotions.
   - If no words are found associated with a particular emotion, that emotion is assigned an intensity value of 0.

3. **Averaging Intensities**:

   - For each emotion, calculate the average intensity of all words associated with that emotion in the text.
   - This is done by summing the intensity scores of all relevant words for a particular emotion and dividing by the number of such words.
   - The resulting average intensity score represents the overall emotional intensity for that emotion within the text.

4. **Output**: The final output for each text is a set of eight emotional intensity scores, one for each of the emotions defined in the NRC-EIL. These scores quantify the intensity of anger, anticipation, disgust, fear, joy, sadness, surprise, and trust expressed in the text.

Code Implementation

```
[ ] # Converting all texts to lower case because lexicon words are in lower case
    df['text'] = df['text'].apply(str.lower)

    emotion_intensity_dict = {emotion: [] for emotion in lexicons_df['lex_emotion'].unique()}

    for _, row in lexicons_df.iterrows():
        emotion_intensity_dict[row['lex_emotion']].append(row['lex_intensity'])

    result_dict_list = []
    for _, row in df.iterrows():
        text = row['text']
        temp_intensity_dict = {emotion: [] for emotion in lexicons_df['lex_emotion'].unique()}

        words = text.split()  # Tokenize the text into words

        # Check if the word is present in lexicons_df and update the temp_intensity_dict
        for word in words:
            matching_rows = lexicons_df[lexicons_df['lex_word'] == word]
            for _, match_row in matching_rows.iterrows():
                temp_intensity_dict[match_row['lex_emotion']].append(match_row['lex_intensity'])

        # Calculate the average intensity for each emotion
        average_intensity_dict = {
            emotion: sum(intensities) / len(intensities) if len(intensities) > 0 else 0
            for emotion, intensities in temp_intensity_dict.items()
        }

        result_dict_list.append(average_intensity_dict)

    result_df = pd.DataFrame(result_dict_list)

    # Fill any missing emotions with 0 intensity
    for emotion in emotion_intensity_dict.keys():
        if emotion not in result_df.columns:
            result_df[emotion] = 0.0

    # Reorder columns based on the original order in lexicons_df
    result_df = result_df[lexicons_df['lex_emotion'].unique()]
    result_df
```

Figure 4.4.3: Emotion Intensity Code Implementation

Example

To demonstrate how emotional intensities are calculated, let's consider the following sentence:

> *"She was so happy initially, but seeing the difficulties of the questions in the exam, she had to cheat and deceive, and was now sad and guilty."*

This sentence contains the following words and their associated emotions from the NRC Emotion Intensity Lexicon, and we calculate the average intensity for each emotion accordingly using Figure 4.1.1.

Joy: happy (0.868)

$$\text{Sum of intensities} = 0.868$$

$$\text{Number of words} = 1$$

$$\text{Average intensity} = \frac{0.868}{1} = 0.868$$

Sadness: sad (0.844), guilt (0.750), difficulties (0.421)

$$\text{Sum of intensities} = 0.844 + 0.750 + 0.421 = 2.015$$

$$\text{Number of words} = 3$$

$$\text{Average intensity} = \frac{2.015}{3} = 0.671$$

Anger: cheat (0.630), deceive (0.547)

$$\text{Sum of intensities} = 0.630 + 0.547 = 1.177$$

$$\text{Number of words} = 2$$

$$\text{Average intensity} = \frac{1.177}{2} = 0.588$$

Note: Since no words are from another emotion, all the other emotions are assigned an intensity of 0.

4.4.4 Emotional Intensity Analysis

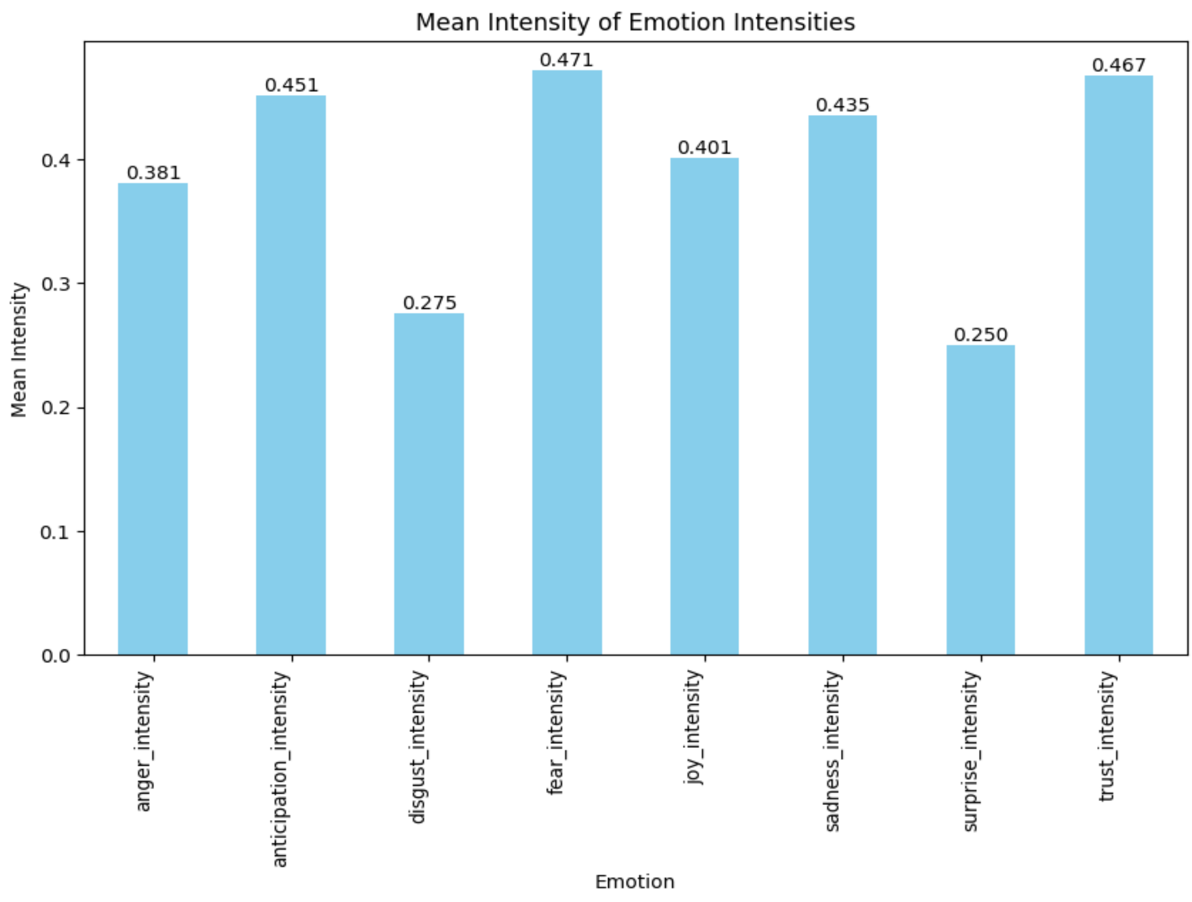


Figure 4.4.4: Average Probabilities of Emotional Intensities for entire dataset

Figure 4.4.4 displays the mean intensity of eight emotional intensities for the entire dataset, revealing that **fear intensity** has the highest mean intensity at **0.471.** This indicates that fear is the most prevalent emotion across the dataset, potentially due to the nature of the subjects or scenarios involved, which might be stress-inducing or anxiety/panic-provoking situations. **Trust**

intensity follows closely with a mean of **0.467,** suggesting that despite the high levels of fear, there is also a significant presence of trust. This could be due to reliance on support systems or coping mechanisms in challenging situations. Other notable intensities include **anticipation (0.451)** and **sadness (0.435),** indicating a balanced intensity distribution between the positive and negative emotions in the texts.

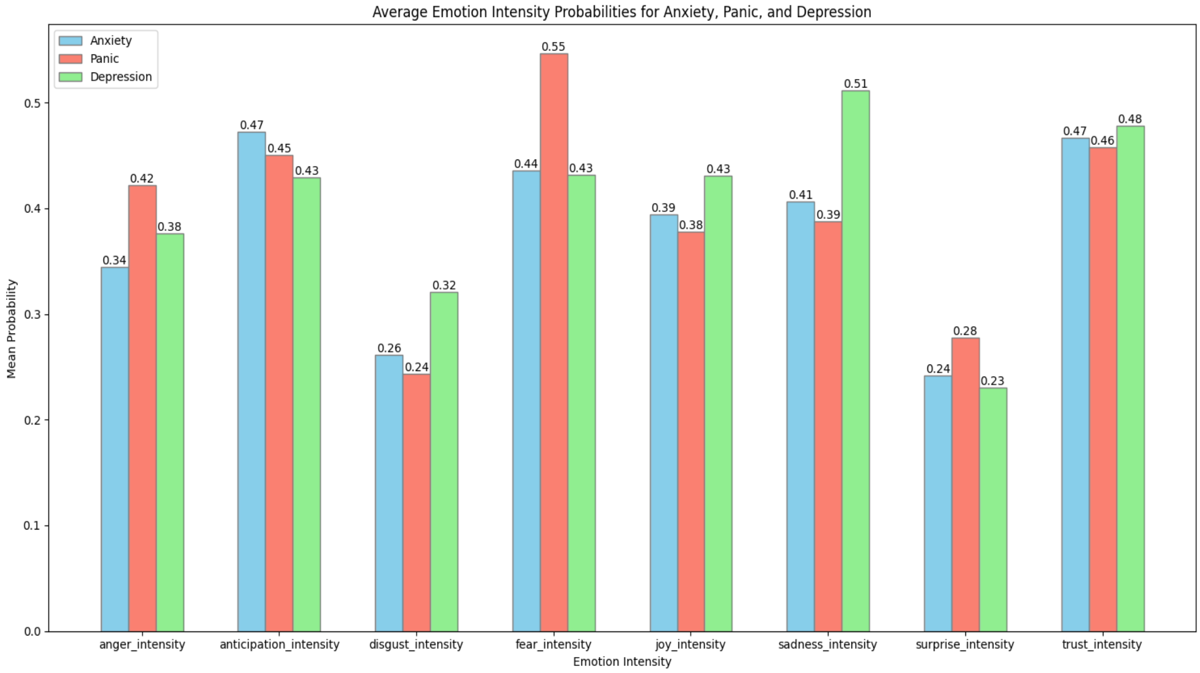


Figure 4.4.5: Average Probabilities of Emotional Intensities for anxiety, panic, and depression datasets

Figure 4.4.5 displays the average of emotional intensities across each label individually. **Anger intensity** is highest in the **panic dataset (0.42),** suggesting that panic attacks might trigger feelings of anger due to sudden loss of control. **Depression** follows with an anger intensity of **0.38**, and **anxiety is at 0.34,** likely linked to frustration and irritability. **Anticipation intensity** is the highest in **anxiety (0.47),** reflecting concerns about future events, followed by **panic (0.45)** and **depression (0.43),** which suggests a lack of optimism. **Disgust intensity,** while submissive overall, is highest in **depression (0.32),** implying self-directed negative feelings, with **anxiety at 0.26** and **panic at 0.24. Fear intensity** peaks in **panic (0.55),** consistent with intense fear during panic attacks, while **anxiety and depression** show high fear levels at **0.44 and 0.43,** respectively. **Joy intensity** is highest in **depression (0.43),** followed by **anxiety (0.39)** and **depression (0.38),** possibly reflecting relief post-attack and indicating moments of joy despite chronic worry. **Sadness intensity** is highest in **depression (0.51),** as expected, with **anxiety (0.41)** and **panic (0.39)** also showing notable sadness, indicating significant impacts on individuals in these states. **Surprise intensity** is lowest among all, with **panic at 0.28, anxiety at 0.24, and depression at 0.23,** reflecting the unpredictable nature of panic attacks and constant alertness in anxiety, while depression shows a general feeling of predictability or numbness. **Trust intensity** is highest in **depression (0.48),** indicating reliance on social support or therapeutic relationships, with **anxiety and panic** close behind at **0.47 and 0.46,** respectively.

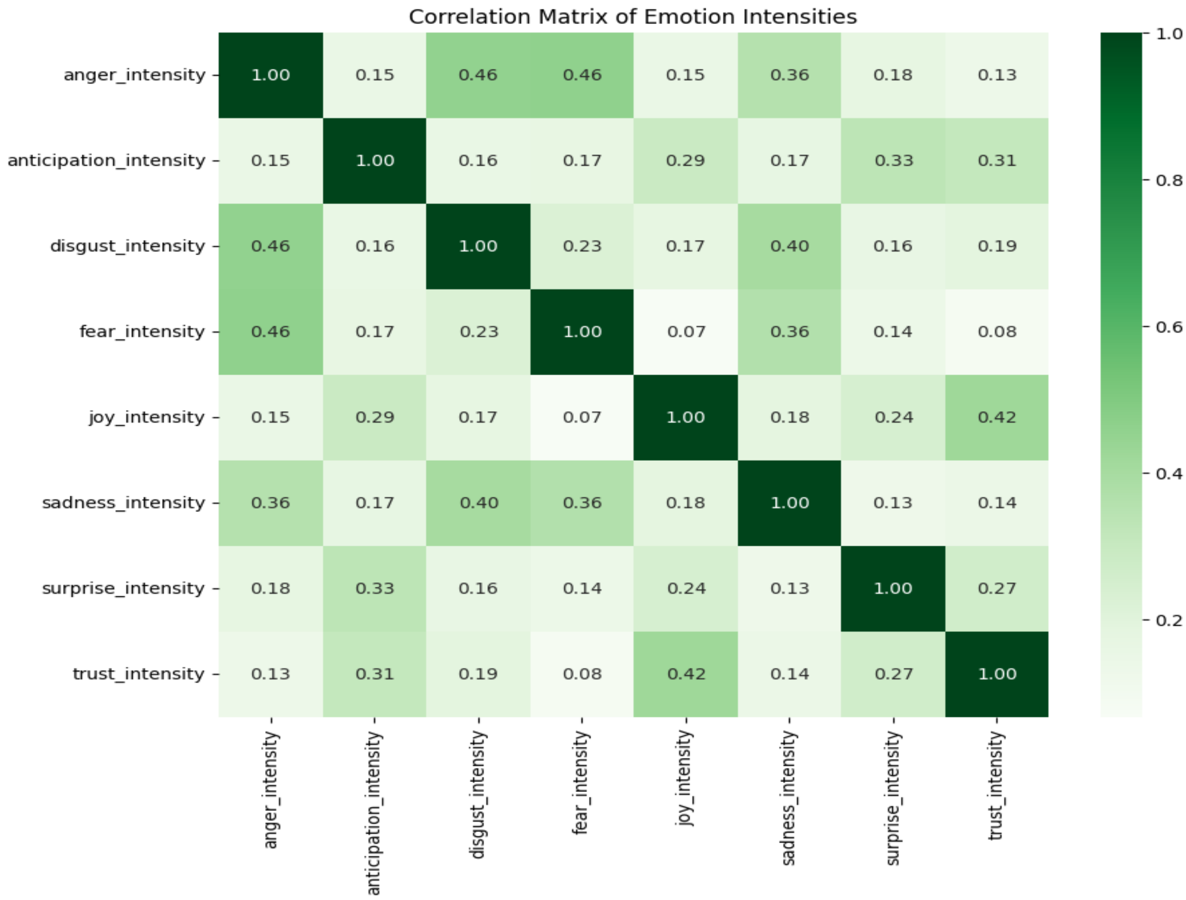


Figure 4.4.6: Correlation Matrix of Emotional Intensities

The correlation matrix (Figure 4.4.6) of the eight emotional intensities shows significant relationships between certain intensities. For instance, **anger intensity and disgust intensity have a high correlation (0.46),** suggesting that these negative emotions often co-occur. Similarly, **joy intensity has a notable correlation with trust intensity (0.42),** indicating that positive emotions are likely to be experienced together. **Low correlations,** such as between **fear intensity and joy intensity (0.07),** reflect the expected inverse relationship where high levels of fear are less likely to coincide with feelings of joy. These correlations provide insights into the complex interplay of emotions, where certain emotions are more likely to be experienced simultaneously than others.

Chapter 5

Methodology

5.1 Model Selection

In this section, we delve into the selection and exploration of diverse models for our classification task. Our choice encompasses Logistic Regression (LR), XGBoost (XGB), and Multi-layered Deep Learning (DL) models. Each model was selected based on its appropriateness and efficacy for the given task, with Logistic Regression serving as the foundational approach, XGBoost leveraging its proven performance, and Deep Learning delving into the intricacies of neural networks. Below, we provide detailed descriptions and the underlying mathematics of these models.

5.1.1 Logistic Regression

Logistic Regression is a statistical method used for classification tasks, where the goal is to predict the probability that an instance belongs to a particular class. For binary classification tasks, the dependent variable is binary, meaning it takes on only two possible outcomes (e.g., 0 or 1, true or false, yes or no). The independent variables can be continuous, categorical, or a mix of both. In logistic regression for multiclass classification, there are different strategies to handle the multiclass nature of the problem, namely One-vs-One (OvO), One-vs-Rest (OvR), and Multinomial approaches. We use the Multinomial approach as it directly extends logistic regression to handle multiple classes in a single model. It uses the softmax function to compute probabilities for each class, ensuring they sum to 1. This method involves optimizing a single objective function that considers all classes simultaneously, rather than training multiple binary classifiers.

The fundamental concept behind Logistic Regression is to model the relationship between the independent variables and the probability of the binary outcome using the logistic function (also known as the sigmoid function). It is a mathematical function used to map the predicted values to probabilities. The logistic function ensures that the predicted probabilities fall within the range of 0 and 1, making it suitable for binary classification problems. The logistic regression model predicts the probability that the dependent variable Y belongs to a particular class given the values of the independent variables X. If the predicted probability is greater than a threshold (usually 0.5), the instance is classified into one class; otherwise, it is classified into the other class.

The logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

where z is a linear combination of the independent variables:

$$z = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n$$

Here, θ_0 is the intercept and $\theta_1, \theta_2, \ldots \theta_n$ are the coefficients of the independent variables $X_1, X_2, \ldots X_n$.

The logistic function converts the linear combination z into a probability value between 0 and 1. The probability p of the dependent variable Y being 1 (true) given the independent variables X is:

$$p(Y = 1|X) = \sigma(z) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n)}}$$

Conversely, the probability of Y being 0 (false) is:

$$p(Y = 0|X) = 1 - p(Y = 1|X) = 1 - \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n)}}$$

The coefficients θ are estimated using Maximum Likelihood Estimation (MLE). In practice, the log-likelihood is used to simplify the calculation. The goal is to find the parameters θ that maximize the log-likelihood function. To convert the maximization problem of the log-likelihood into a minimization problem (more convenient for optimization), we take the negative log-likelihood:

$$-\log L(\theta) = -\left[y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))\right]$$

The cost function for Logistic Regression is derived from the likelihood function, specifically the negative log-likelihood. This approach is used because the logistic regression model estimates probabilities, and the log-likelihood is a measure of how well the model predicts the observed outcomes.

Given a set of training data with m examples, where y_i is the actual label for the i-th example and $h_\theta(x_i)$ is the predicted probability that $y_i = 1$, the cost function $J(\theta)$ is defined as:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))\right]$$

The goal of training a Logistic Regression model is to find the parameters θ that minimize the cost function $J(\theta)$. This is typically done using optimization algorithms such as Gradient Descent, where parameters are updated iteratively in the direction of the negative gradient of the cost function:

$$\theta := \theta - \alpha \nabla_\theta J(\theta)$$

Here, α is the learning rate, and $\nabla_\theta J(\theta)$ is the gradient of the cost function with respect to the parameters θ.

In summary, Logistic Regression is a powerful and widely used tool for binary classification tasks. Its ability to output probabilities, handle non-linear relationships, and maintain interpretability makes it a preferred choice over Linear Regression for many practical applications.

5.1.2 Extreme Gradient Boosting (XGB)

Overview of XGBoost

XGBoost, short for Extreme Gradient Boosting, is an efficient and scalable implementation of the gradient boosting framework. It has become widely popular in machine learning competitions (such as Kaggle Competitions) and real-world applications due to its robust performance and efficiency. Here are some key reasons why XGBoost performs exceptionally well for classification tasks:

- Regularization: XGBoost includes regularization parameters to avoid overfitting, making the model more generalizable.
- Handling Missing Values: It has a built-in mechanism to handle missing values, which can be a common issue in real-world datasets.
- Parallel Processing: XGBoost can make use of multiple cores on a CPU to perform parallel processing, speeding up training time.
- Tree Pruning: It uses a depth-first approach and sophisticated tree pruning to handle overfitting and improve model accuracy.
- Sparsity Awareness: The algorithm is optimized for sparse data and can handle large datasets with high-dimensional features efficiently.

Mathematical Foundations of XGBoost
XGBoost uses an ensemble of decision trees to make predictions. It works by sequentially adding simple models to correct the errors made by previous models. Following are the mathematical components of the model.

Hypothesis Function
In XGBoost, the prediction for a given data point is made by summing the predictions of all individual regression trees in the ensemble. If there are K trees, the prediction for the i_{th} data point $\mathbf{x}_i$ is given by:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i)$$

where each f_k represents a decision tree in the ensemble.

Objective Function
The objective function in XGBoost consists of two parts: the loss function and the regularization term. The goal is to minimize the overall objective function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where l is the loss function measuring the difference between the predicted value $\hat{y}_i$ and the true value y_i, and Ω is the regularization term for the complexity of the model.

Loss Function
For a classification task, a common choice for the loss function is the logistic loss:

$$l(y, \hat{y}) = -[y \log(\sigma(\hat{y})) + (1 - y) \log(1 - \sigma(\hat{y}))]$$

where $\sigma(\hat{y})$ is the sigmoid function defined as $\sigma(\hat{y}) = \frac{1}{1+\exp(-\hat{y})}$.

Regularization Term
The regularization term Ω controls the complexity of the model and helps prevent overfitting. It is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

where T is the number of leaves in the tree, w_j is the weight of leaf j, γ is a parameter controlling the complexity of the model (penalizing the number of leaves), and λ is the L2 regularization term on leaf weights.

Additive Training
XGBoost uses an additive approach to minimize the objective function. At each iteration t, a new function f_t is added to the model to minimize the objective:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

Taylor Expansion for Optimization
To efficiently optimize the objective function, XGBoost uses a second-order Taylor expansion approximation:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t)$$

where g_i and h_i are the first and second derivatives of the loss function with respect to the prediction $\hat{y}_i^{(t-1)}$.

Similarity Weights
Similarity weights in XGBoost help in determining the optimal splits in the decision tree. These weights are computed based on the residuals, which are the differences between the observed and predicted values. For a given leaf j, the similarity weight S_j is calculated as:

$$S_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

where: - $\sum_{i \in I_j} g_i$ is the sum of the residuals (first-order gradients) of the instances in the leaf. - $\sum_{i \in I_j} h_i$ is the sum of the second-order gradients of the instances in the leaf. - λ is the regularization term.

Information Gain
Information gain measures the improvement in the objective function from splitting a node into two leaves. It is calculated as the difference in the similarity score before and after the split. The gain from splitting a node into two leaves is given by:

$$\text{Gain} = \frac{1}{2} \left(\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I_L \cup I_R} g_i)^2}{\sum_{i \in I_L \cup I_R} h_i + \lambda} \right) - \gamma$$

where: - $\sum_{i \in I_L} g_i$ and $\sum_{i \in I_R} g_i$ are the sums of the residuals (first-order gradients) for the left and right leaves, respectively. - $\sum_{i \in I_L} h_i$ and $\sum_{i \in I_R} h_i$ are the sums of the second-order gradients for the left and right leaves, respectively. - $\sum_{i \in I_L \cup I_R} g_i$ is the sum of the residuals for the entire node before the split. - $\sum_{i \in I_L \cup I_R} h_i$ is the sum of the second-order gradients for the entire node before the split. - γ is the regularization term for the number of leaves.
The overall gain from splitting a node into two leaves is calculated, and the split that provides the highest gain is chosen.

Summary
XGBoost's efficiency and performance stem from its ability to handle complex data structures with regularization, robust handling of missing values, and parallel processing capabilities. The mathematical foundation, including the hypothesis function, loss function, regularization,

optimization through Taylor expansion, similarity weights, and information gain based on residuals, makes it a powerful tool for classification tasks.

5.1.3 Multi-layered Deep Learning Architecture

Overview of Neural Networks
Neural networks are computational models inspired by the structure and function of the human brain. They are designed to recognize patterns, make predictions, and solve complex problems by learning from data. The fundamental building block of a neural network is the neuron, which is an individual computational unit. Neurons are organized into layers, typically including an input layer, one or more hidden layers, and an output layer.

Neurons, Weights, and Biases: Each neuron receives inputs, which can be features from the data or outputs from other neurons in the previous layer. These inputs are each associated with a weight, which determines the importance or influence of that input on the neuron's output. The neuron computes a weighted sum of its inputs and adds a bias term. The bias is a constant value that allows the neuron to better fit the data by shifting the activation function. The activation function is then applied to this sum to introduce non-linearity into the model, enabling the network to learn complex patterns. Mathematically, this process is expressed as:

$$\text{output} = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

where w_i are the weights, x_i are the inputs, b is the bias, and σ is the activation function (e.g., sigmoid, ReLU).

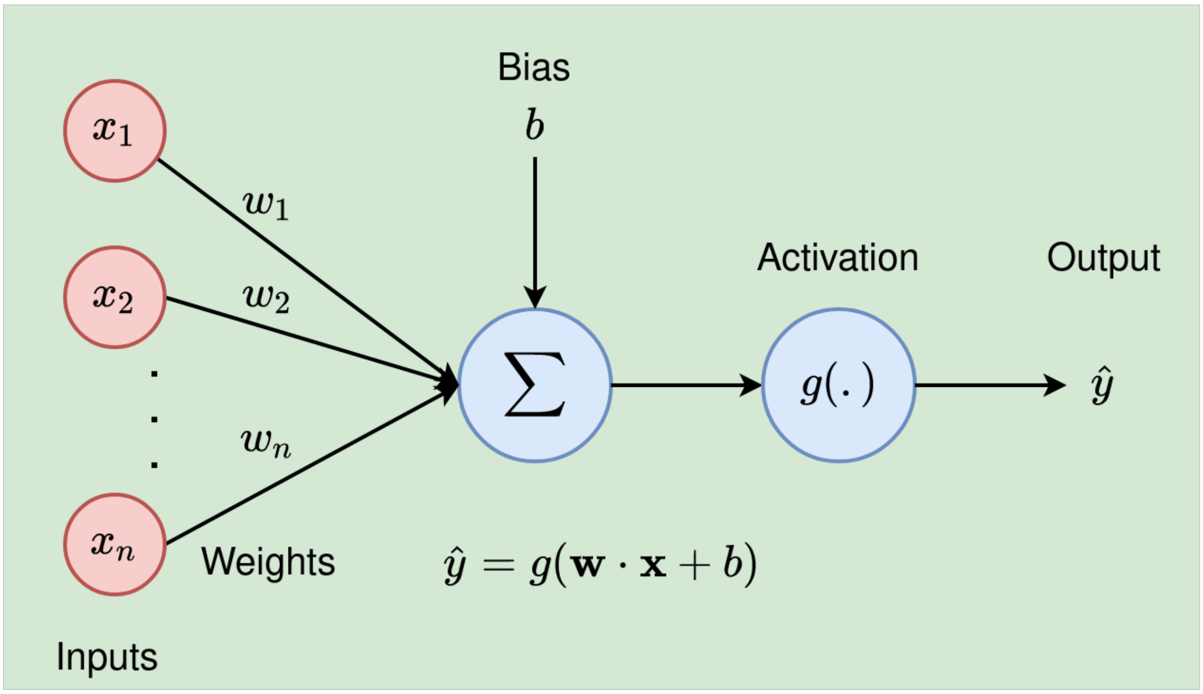


Figure 5.1.1: Working of Neural Networks

Hidden Layers: Between the input and output layers, neural networks often have one or more hidden layers. These hidden layers consist of neurons that perform intermediate computations, transforming the input into useful representations. Each hidden layer learns different features of the data, with deeper layers typically learning more abstract features. The introduction of hidden layers allows the network to model complex relationships and interactions between the inputs, which would be impossible with a single layer.

Output Layer: The final layer in a neural network is the output layer, which produces the model's predictions. The number of neurons in the output layer corresponds to the number of desired outputs. For instance, in a classification problem with three classes (A, B, C), the output layer would have three neurons. The activation function used in the output layer depends on the task. For binary classification, a sigmoid function is commonly used, while for multiclass classification, a softmax function is used to produce a probability distribution over the classes.

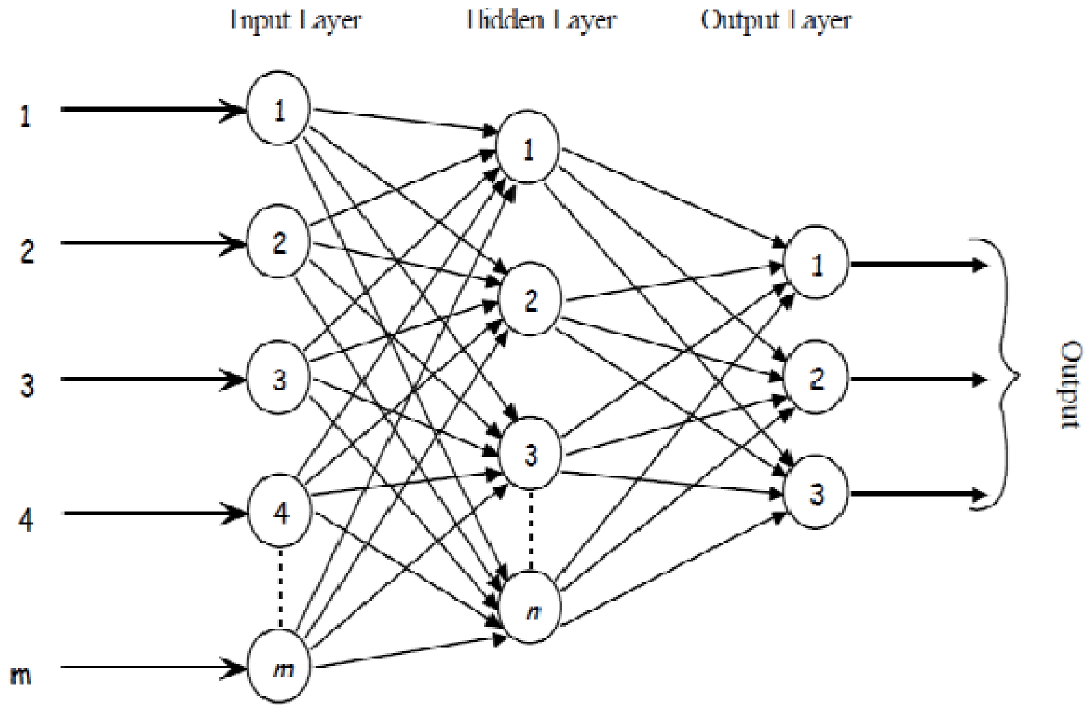


Figure 5.1.2: Simple Neural Network Architecture

Backpropagation: Backpropagation is the algorithm used to train neural networks. It involves a forward pass, where the input data is passed through the network to generate an output, and a backward pass, where the error between the predicted and actual output is propagated back through the network. During the backward pass, the algorithm computes the gradient of the error with respect to each weight and bias in the network. These gradients are used to update the weights and biases using gradient descent or other optimization techniques, with the goal of minimizing the error.

Long Short-Term Memory (LSTM)
Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to handle sequential data and capture long-term dependencies. Unlike traditional RNNs, LSTMs can effectively learn and remember over long sequences, making them particularly useful for tasks involving time series data, natural language processing, and any other domain where the order of inputs is important.
Application in Texts: LSTM layers are beneficial for text data as they can retain context and semantic meaning across long sentences and paragraphs. This capability allows them

to understand the relationship between words and phrases, making them ideal for tasks like sentiment analysis, language translation, and speech recognition.

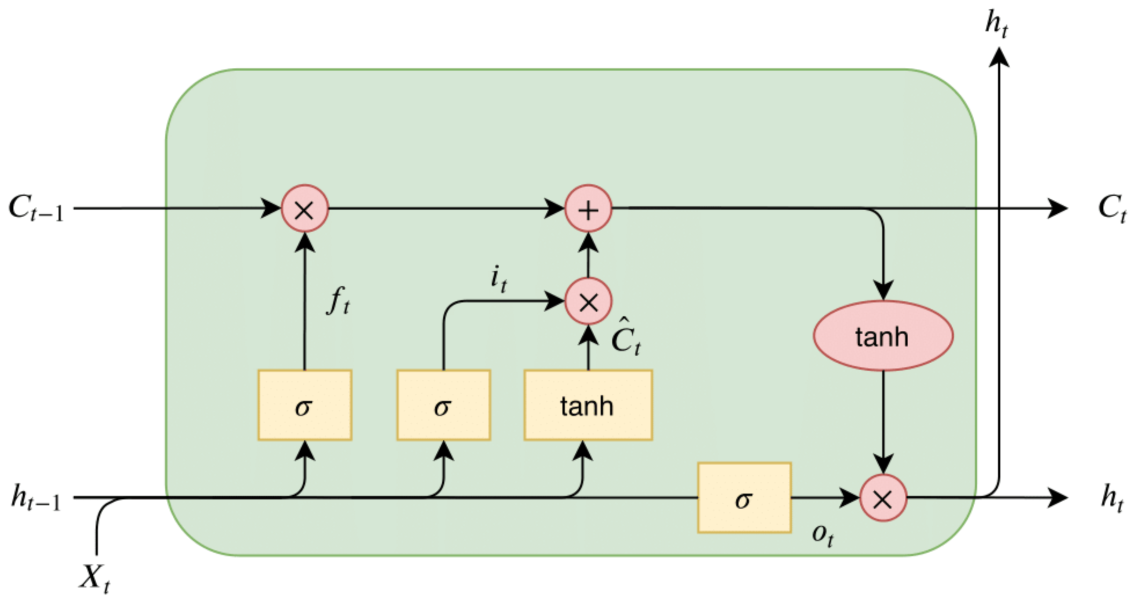


Figure 5.1.3: LSTM architecture

Convolutional Neural Networks (CNN)
Convolutional Neural Networks (CNNs) are specialized neural networks designed to process data with grid-like topology, such as images. They utilize convolutional layers to automatically and adaptively learn spatial hierarchies of features from input images. CNNs apply convolution operations followed by pooling operations to reduce the dimensionality of the data, focusing on the most relevant features.
Application in Texts: While CNNs are primarily known for image processing, they can also be applied to text data, especially for tasks like sentence classification and sentiment analysis. By treating text as a one-dimensional grid, CNNs can learn local patterns such as n-grams, which capture the local structure of text data. This allows the network to understand the context and semantics of words within their local neighborhoods.

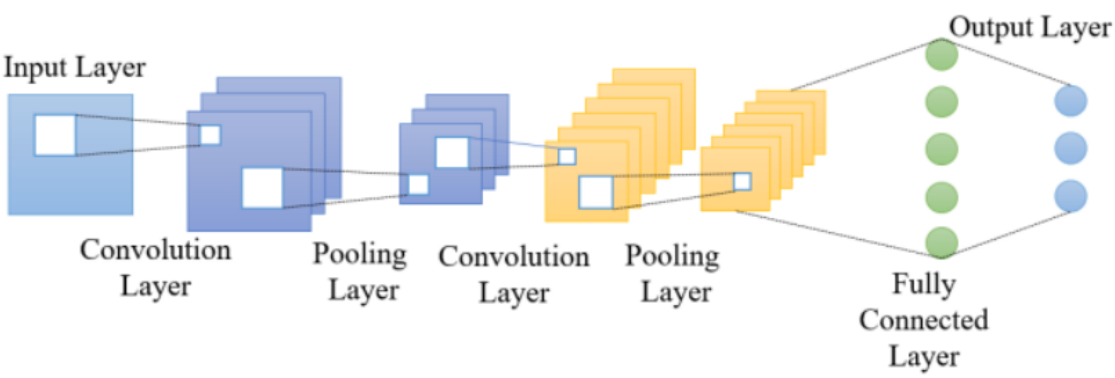


Figure 5.1.4: CNN architecture

Working of Multi-layered Deep Learning Architecture
In this work, a multi-layered deep learning architecture is employed [34], where different features of the dataset are processed using different types of layers. Specifically:

- An LSTM layer is applied to one feature, leveraging its ability to capture sequential dependencies.
- A CNN layer is applied to another feature, exploiting its capability to learn local patterns.
- The remaining two features are processed using a simple neural network architecture without additional layers.

This approach allows the model to extract and combine various types of information from the features, potentially leading to better performance on the task at hand.

5.2 Training method and Evaluation Metrics

5.2.1 10-Fold Cross Validation

10-fold cross-validation is a robust technique for evaluating the performance of a machine learning model. It involves partitioning the original dataset into 10 equal-sized subsets, or "folds." The model is then trained and evaluated 10 times, each time using a different fold as the test set and the remaining 9 folds as the training set. The model's performance is evaluated on the test set using various metrics, and the results from the 10 iterations are averaged to produce a single estimate of model performance.

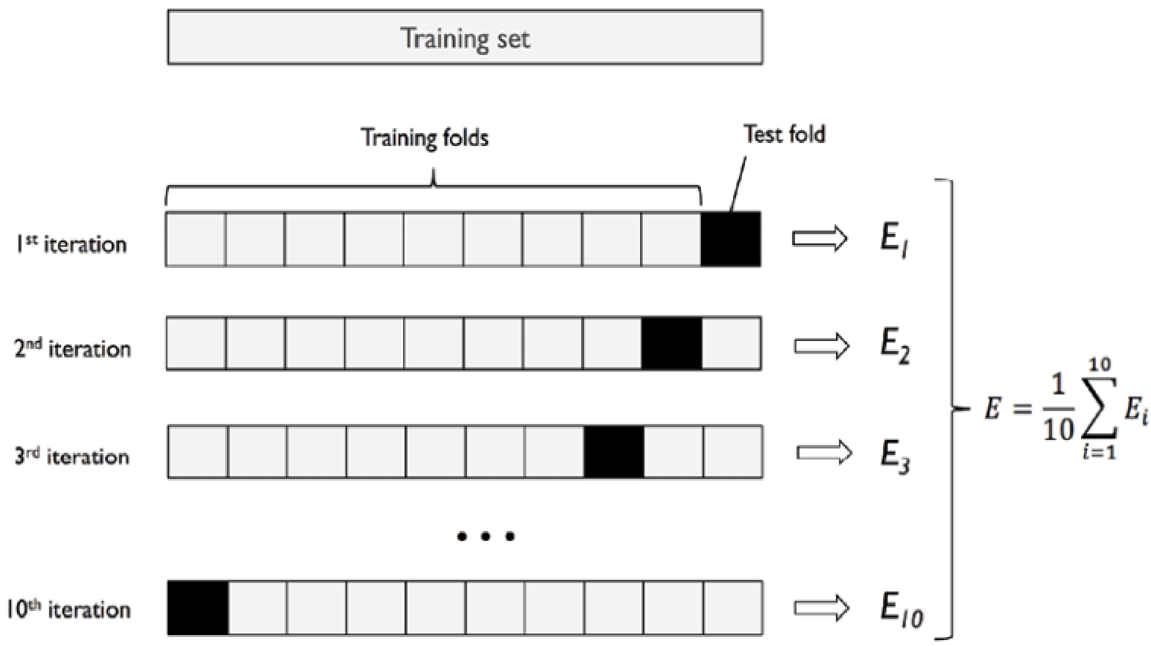


Figure 5.2.1: Working of 10-fold Cross Validation

This process ensures that every data point is used for both training and testing, providing a comprehensive assessment of the model's ability to generalize to unseen data.

5.2.2 Evaluation Metrics for Multi-class Classification

We employed several evaluation metrics for evaluating our Logistic Regression and Machine Learning models such as **F1 score, Micro F1 score, Macro F1 score, and ROC AUC Score.**

In the context of evaluating classification models, the terms True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are crucial for understanding the performance of the model.

- TP (True Positive): The number of correctly predicted positive labels.
- TN (True Negative): The number of correctly predicted negative labels.
- FP (False Positive): The number of incorrectly predicted positive labels.
- FN (False Negative): The number of incorrectly predicted negative labels.

Let's consider a multi-class classification problem with three classes: A, B, and C.

Precision, Recall, and F1 Score
For each class, the precision, recall, and F1 score are calculated individually. Consider one of the classes, say class A.

- **Precision**: Precision is the ratio of true positive predictions to the total predicted positives. It measures the accuracy of the positive predictions for a particular class.

$$\text{Precision}_A = \frac{\text{True Positives}_A}{\text{True Positives}_A + \text{False Positives}_A} \tag{5.1}$$

- **Recall**: Recall is the ratio of true positive predictions to the total actual positives. It measures the ability of the model to identify all positive instances for a particular class.

$$\text{Recall}_A = \frac{\text{True Positives}_A}{\text{True Positives}_A + \text{False Negatives}_A} \tag{5.2}$$

- **F1 Score**: The F1 score is the harmonic mean of precision and recall, providing a balance between the two for a particular class.

$$\text{F1 Score}_A = 2 \times \frac{\text{Precision}_A \times \text{Recall}_A}{\text{Precision}_A + \text{Recall}_A} \tag{5.3}$$

Similarly, the precision, precision, and f1 score of classes B and C are calculated individually.

Macro F1 Score
The macro F1 score is the average of the F1 scores for each class. It treats all classes equally, regardless of their support (number of true instances).

$$\text{Macro F1} = \frac{1}{3}(\text{F1 Score}_A + \text{F1 Score}_B + \text{F1 Score}_C) \tag{5.4}$$

Micro F1 Score
The micro F1 score aggregates the contributions of all classes to compute the average metric. It is calculated by globally considering the total true positives, false positives, and false negatives.

- **Micro Precision and Micro Recall**: These are calculated by summing the true positives, false positives, and false negatives across all classes and then computing precision and recall.

$$\text{Micro Precision} = \frac{\sum \text{True Positives}}{\sum \text{True Positives} + \sum \text{False Positives}} \tag{5.5}$$

$$\text{Micro Recall} = \frac{\sum \text{True Positives}}{\sum \text{True Positives} + \sum \text{False Negatives}} \tag{5.6}$$

- **Micro F1 Score**: The micro F1 score is then calculated using the micro precision and micro recall.

$$\text{Micro F1} = 2 \times \frac{\text{Micro Precision} \times \text{Micro Recall}}{\text{Micro Precision} + \text{Micro Recall}} \tag{5.7}$$

ROC AUC Score
The ROC AUC (Receiver Operating Characteristic Area Under the Curve) score measures the model's ability to distinguish between classes. For multiclass classification, the one-vs-rest (OvR) approach is typically used, which involves calculating the ROC AUC score for each class against all other classes and then averaging the scores. The ROC curve plots the true positive rate (recall) against the false positive rate (1 - specificity) at various threshold settings. A higher ROC AUC score (closer to 1) indicates that the model has better discrimination ability, i.e., it correctly ranks the positive and negative instances more effectively.

For each class, the ROC AUC score is calculated as:

$$\text{ROC AUC}_A = \text{AUC of ROC Curve for class A vs. not A} \tag{5.8}$$

$$\text{ROC AUC}_B = \text{AUC of ROC Curve for class B vs. not B} \tag{5.9}$$

$$\text{ROC AUC}_C = \text{AUC of ROC Curve for class C vs. not C} \tag{5.10}$$

The average ROC AUC score for the multiclass classification is the average of these individual ROC AUC scores.

For our multi-layered deep learning model, along with F1 score, we employed Accuracy, Sparse Categorical Cross-Entropy loss, and Matthews Correlation Coefficient (MCC).

Accuracy
Accuracy is the ratio of correctly predicted instances to the total instances. It measures how often the model's predictions are correct overall.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Sparse Categorical Cross-Entropy
Sparse Categorical Cross-Entropy loss is used for multi-class classification where the target variable is an integer representing the class. It measures the difference between the true labels and the predicted probabilities.

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{p}_{y_i})$$

Where N is the number of samples, y_i is the true label of the i-th sample, and $\hat{p}_{y_i}$ is the predicted probability of the true class y_i.

Matthews Correlation Coefficient (MCC)
MCC is a correlation coefficient between the observed and predicted classifications, which returns a value between -1 and +1. An MCC of +1 indicates perfect prediction, 0 indicates no better than random prediction, and -1 indicates total disagreement between prediction and observation.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

These metrics provide a comprehensive evaluation of our model's performance, taking into account not only the accuracy of predictions but also the probabilistic confidence and the balance between different classes.

5.2.3 Code Implementation of the models

Logistic Regression and XGBoost
The code implementation of Logistic Regression and XGBoost, using 10-fold cross-validation, and assessed using the above evaluation metrics is shown in Figure 5.2.2.

```python
from sklearn.model_selection import StratifiedKFold
folds = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

clf = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=20000)
clf_xgb = xgb.XGBClassifier(objective='multi:softmax', missing=0, seed=42, num_class=3)

logistic_f1scores = []
logistic_f1microscores = []
logistic_f1macroscores = []
logistic_rocaucscore = []
xgb_f1scores = []
xgb_f1microscores = []
xgb_f1macroscores = []
xgb_rocaucscore = []

for train_index, test_index in folds.split(X, y):
    X_train, X_test, y_train, y_test = X[train_index], X[test_index], y[train_index], y[test_index]

    # Logistic Regression
    clf.fit(X_train, y_train)
    y_pred_logistic = clf.predict(X_test)
    y_pred_proba_logistic = clf.predict_proba(X_test)
    f1_logistic = f1_score(y_test, y_pred_logistic, average='weighted')
    f1_micro_logistic = f1_score(y_test, y_pred_logistic, average='micro')
    f1_macro_logistic = f1_score(y_test, y_pred_logistic, average='macro')
    logistic_f1scores.append(f1_logistic)
    logistic_f1microscores.append(f1_micro_logistic)
    logistic_f1macroscores.append(f1_macro_logistic)
    rocauc_logistic = roc_auc_score(y_test, y_pred_proba_logistic, multi_class='ovr')
    logistic_rocaucscore.append(rocauc_logistic)

    # XGBoost
    clf_xgb.fit(X_train, y_train,
              verbose=False,  # set to True for individual use
              early_stopping_rounds=10,
              eval_metric='mlogloss',
              eval_set = [(X_test,y_test)])
    y_pred_xgb = clf_xgb.predict(X_test)
    y_pred_proba_xgb = clf_xgb.predict_proba(X_test)
    f1_xgb = f1_score(y_test, y_pred_xgb, average='weighted')
    f1_micro_xgb = f1_score(y_test, y_pred_xgb, average='micro')
    f1_macro_xgb = f1_score(y_test, y_pred_xgb, average='macro')
    xgb_f1scores.append(f1_xgb)
    xgb_f1microscores.append(f1_micro_xgb)
    xgb_f1macroscores.append(f1_macro_xgb)
    rocauc_xgb = roc_auc_score(y_test, y_pred_proba_xgb, multi_class='ovr')
    xgb_rocaucscore.append(rocauc_xgb)
```

Figure 5.2.2: Code Implementation of Logistic Regression and XGBoost

Multi-layered Deep Learning model

The code implementation of the Multi-layered Deep Learning model, using 10-fold cross-validation, and assessed using the above evaluation metrics is shown in Figure 5.2.3.

```
for train_index, test_index in folds.split(final_df, y):
    X_sentemb_train, X_sentemb_test = X_sentemb.iloc[train_index], X_sentemb.iloc[test_index]
    X_emotions_train, X_emotions_test = X_emotions.iloc[train_index], X_emotions.iloc[test_index]
    X_intensity_train, X_intensity_test = X_intensity.iloc[train_index], X_intensity.iloc[test_index]
    X_liwc_train, X_liwc_test = X_liwc.iloc[train_index], X_liwc.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    input_sentemb = Input(shape=(X_sentemb.shape[1],))
    lstm_sentemb = LSTM(32)(Reshape((1, len(X_sentemb.columns)))(input_sentemb))

    input_emotions = Input(shape=(X_emotions.shape[1],))  # (28,)
    reshaped_emotions = Reshape((X_emotions.shape[1], 1))(input_emotions)
    cnn_emotions = Conv1D(128, 3, activation='relu')(reshaped_emotions)
    cnn_emotions = MaxPooling1D(pool_size=5, strides=2)(cnn_emotions)
    cnn_emotions = Flatten()(cnn_emotions)
    # + LSTM
    #cnn_lstm_emotions = LSTM(32)(Reshape((1, cnn_emotions.shape[1]))(cnn_emotions))

    input_intensity = Input(shape=(X_intensity.shape[1],))
    #lstm_intensity = LSTM(32)(Reshape((1, len(X_intensity.columns)))(input_intensity))

    input_liwc = Input(shape=(X_liwc.shape[1],))

    # Concatenate the outputs of the LSTM and CNN layers
    concatenated = Concatenate()([lstm_sentemb, cnn_emotions, input_intensity, input_liwc])
    merged_output = Dense(512, activation='relu')(concatenated)
    merged_output = Dropout(rate=0.2)(merged_output)
    merged_output = Dense(256, activation='relu')(concatenated)
    merged_output = Dropout(rate=0.2)(merged_output)
    merged_output = Dense(128, activation='relu')(concatenated)
    merged_output = Dropout(rate=0.2)(merged_output)
    merged_output = Dense(64, activation='relu')(merged_output)
    merged_output = Dropout(rate=0.2)(merged_output)
    merged_output = Dense(32, activation='relu')(merged_output)
    merged_output = Dropout(rate=0.2)(merged_output)
    output = Dense(3, activation='softmax')(merged_output)

    model = Model(inputs=[input_sentemb, input_emotions, input_intensity, input_liwc], outputs=output)
    early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy', F1Score()])
    model.fit([X_sentemb_train, X_emotions_train, X_intensity_train, X_liwc_train], y_train,
              epochs=20, batch_size=64, validation_data=([X_sentemb_test, X_emotions_test, X_intensity_test,
                                                          X_liwc_test], y_test), callbacks=[early_stopping])
    loss, accuracy, f1 = model.evaluate([X_sentemb_test, X_emotions_test, X_intensity_test, X_liwc_test], y_test)
    y_pred = model.predict([X_sentemb_test, X_emotions_test, X_intensity_test, X_liwc_test])
    y_pred_classes = np.argmax(y_pred, axis=1)
    mcc = matthews_corrcoef(y_test, y_pred_classes)
```

Figure 5.2.3: Code Implementation of Multi-layered Deep Learning model

Model parameters: All the CNN layers were applied with 128 filters, using a kernel size of 3 and "relu" activation function. After applying LSTM/CNN layers to various features and concatenating them into the input layer, we constructed 5 hidden layers with 512, 256, 128, 64, and 32 neurons, each with a dropout rate of 0.2. Finally, an output layer with 3 neurons and a "softmax" activation function was added. EarlyStopping was implemented with the parameter monitor='val_loss', patience=5, restore_best_weights=True. The model was compiled with loss='sparse_categorical_crossentropy' and optimizer='adam'.

Chapter 6

Results and Discussion

6.1 Experiments and Results

6.1.1 Sentence Embeddings Results

Selection of Embedding Models
Given the variety of models available through HuggingFace, it's crucial to evaluate multiple options to identify the one that best suits the specific characteristics and requirements of our dataset. For this purpose, we experimented with the following four models:

- **bert-base-nli-mean-tokens:** A model based on BERT (Bidirectional Encoder Representations from Transformers) fine-tuned for natural language inference tasks, offering robust contextual embeddings.
- **distilbert-base-nli-mean-tokens:** A distilled version of BERT, providing a lighter and faster alternative while retaining a significant portion of BERT's performance.
- **bert-base-uncased:** The uncased variant of BERT, which disregards case sensitivity and is widely used for a variety of NLP tasks.
- **all-distilroberta-v1:** Based on DistilRoBERTa, a distilled version of RoBERTa (A Robustly Optimized BERT Approach), known for its efficiency and performance in sentence embeddings.

Experimental Evaluation
Each of these models was employed to generate sentence embeddings for our dataset. Subsequently, these embeddings were used as inputs for logistic regression and XGBoost models to assess their effectiveness in classifying texts as anxiety, panic, or depression. The performance focused on evaluation metrics such as F1 score, micro F1 score, macro F1 score, and ROC AUC curve.
The results are presented in Table 6.1.

	LR				XGB			
	Weighted F1	**Micro F1**	**Macro F1**	**ROC AUC Score**	**Weighted F1**	**Micro F1**	**Macro F1**	**ROC AUC Score**
bert-base-nli-mean-tokens	0.729/0.023	0.729/0.024	0.729/0.023	0.882/0.012	0.697/0.020	0.699/0.020	0.697/0.020	0.862/0.014
distilbert-base-nli-mean-tokens	0.725/0.014	0.726/0.014	0.725/0.014	0.881/0.010	0.704/0.028	0.705/0.027	0.704/0.028	0.864/0.020
bert-base-uncased	0.759/0.017	0.760/0.017	0.759/0.017	0.897/0.009	0.720/0.030	0.722/0.029	0.720/0.030	0.878/0.013
all-distilroberta-v1	**0.819/0.020**	**0.820/0.020**	**0.819/0.020**	**0.941/0.011**	**0.801/0.023**	**0.802/0.023**	**0.801/0.023**	**0.931/0.014**

Table 6.1: Evaluation of various Sentence Embeddings Models

Among the models tested, **all-distilroberta-v1** yielded the best results in terms of classification performance, with a **weighted F1 score of 0.819 for Logistic Regression** and **0.801 for XGBoost,** making it the chosen model for subsequent experiments. This model's superior ability to capture the nuances of short biomedical texts contributed significantly to the accuracy and reliability of the mental disorder classification task.

6.1.2 Logistic Regression and XGBoost Results

Here, we present the results of Logistic Regression (LR) and Extreme Gradient Boosting (XGB) by running several experiments with combinations of our four features: sentence embeddings, LIWC, emotions, and intensity. Each experiment was evaluated using Weighted F1, Micro F1, Macro F1, and ROC AUC scores. All values are reported as the mean and standard deviation from 10-fold cross-validation. We normalized our sentence embeddings to ensure that all values fall between 0 and 1, maintaining consistency with the emotions and intensity features. The results are presented in Table 6.2.

	LR				XGB			
	Weighted F1	**Micro F1**	**Macro F1**	**ROC AUC Score**	**Weighted F1**	**Micro F1**	**Macro F1**	**ROC AUC Score**
Emb (all-distilroberta-v1)	0.815/0.018	0.816/0.018	0.815/0.018	0.939/0.012	0.804/0.018	0.804/0.018	0.804/0.018	0.932/0.012
LIWC	0.718/0.020	0.721/0.020	0.718/0.020	0.874/0.015	0.727/0.022	0.733/0.023	0.727/0.022	0.887/0.015
Emotions	0.548/0.014	0.558/0.016	0.547/0.014	0.740/0.009	0.610/0.023	0.615/0.021	0.609/0.023	0.794/0.014
Intensity	0.553/0.016	0.558/0.017	0.552/0.016	0.731/0.011	0.585/0.019	0.587/0.019	0.585/0.019	0.770/0.014
Emb + LIWC	0.807/0.024	0.807/0.024	0.807/0.024	0.933/0.013	0.804/0.022	0.805/0.021	0.804/0.022	0.934/0.013
Emb + Emotions	**0.819/0.020**	**0.819/0.019**	**0.819/0.020**	**0.940/0.011**	**0.801/0.021**	**0.801/0.021**	**0.801/0.021**	**0.932/0.012**
Emb + Intensity	**0.818/0.021**	**0.819/0.021**	**0.818/0.021**	**0.941/0.011**	**0.802/0.020**	**0.802/0.020**	**0.802/0.020**	**0.932/0.011**
Emb + Emotions + Intensity	**0.818/0.017**	**0.819/0.017**	**0.818/0.017**	**0.940/0.011**	**0.797/0.023**	**0.798/0.023**	**0.797/0.023**	**0.931/0.014**
Emb + LIWC + Emotions + Intensity	0.807/0.022	0.807/0.022	0.807/0.022	0.932/0.013	0.805/0.023	0.805/0.023	0.805/0.023	0.935/0.014
Emb + Standardized LIWC	0.809/0.022	0.809/0.022	0.809/0.022	0.934/0.013	0.798/0.025	0.799/0.025	0.798/0.025	0.933/0.011
Emb + Normalized LIWC	0.814/0.021	0.814/0.021	0.814/0.021	0.938/0.012	0.806/0.021	0.807/0.020	0.806/0.021	0.933/0.014
Emb + NorLIWC + Emotions	**0.813/0.020**	**0.813/0.020**	**0.813/0.020**	**0.937/0.011**	**0.803/0.026**	**0.804/0.025**	**0.803/0.026**	**0.935/0.014**
Emb + NorLIWC + Emotions + Intensity	0.812/0.018	0.813/0.017	0.812/0.018	0.937/0.012	0.803/0.022	0.804/0.022	0.803/0.022	0.935/0.014

Table 6.2: Results for LR and XGB

General Performance Overview

- **Sentence Embeddings:** Sentence embeddings alone achieved high-performance metrics across both LR and XGB. For example, using LR, the Weighted F1 score was 0.815 ($\pm$0.018) and the ROC AUC score was 0.939 ($\pm$0.012). Similarly, for XGB, the Weighted F1 score was 0.804 ($\pm$0.018) and the ROC AUC score was 0.932 ($\pm$0.012).

- **Individual Feature Performance:** LIWC, emotions, and intensity, when used individually, did not perform as well as sentence embeddings. For instance, LIWC alone had a Weighted F1 score of 0.718 ($\pm$0.020) with LR and 0.727 ($\pm$0.022) with XGB. Emotions alone had particularly low performance, with a Weighted F1 score of 0.548 ($\pm$0.014) for LR and 0.610 ($\pm$0.023) for XGB. Intensity had an even lower performance with a Weighted F1 score of 0.553 ($\pm$0.016) for LR and 0.585 ($\pm$0.019) for XGB.

- **Integration with Sentence Embeddings:** Integrating LIWC, emotions, and intensity with sentence embeddings significantly improved performance. For example, `Emb + Emotions` achieved a Weighted F1 score of 0.819 ($\pm$0.020) and a ROC AUC score of 0.940 ($\pm$0.011) with LR. Similar improvements were observed with `Emb + Intensity` and `Emb + Emotions + Intensity`, demonstrating that combining these features with sentence embeddings is beneficial.

Experimentation with LIWC Features

- **Normalized and Standardized LIWC:** To maintain similar scaling among all LIWC features, we experimented with normalized and standardized LIWC features. `Emb + Standardized`

`LIWC` showed a slight improvement over `Emb + LIWC`, with a Weighted F1 score of 0.809 (±0.022) for LR compared to 0.807 (±0.024) for `Emb + LIWC`. `Emb + Normalized LIWC` performed even better, achieving a Weighted F1 score of 0.814 (±0.021) and a ROC AUC score of 0.938 (±0.012) with LR.

- **Additional Experiments with Normalized LIWC:** Building on the success of `Emb + Normalized LIWC`, we also experimented with:
  - `Emb + NorLIWC + Emotions`: Achieved a Weighted F1 score of 0.813 (±0.020) and a ROC AUC score of 0.937 (±0.011) with LR.
  - `Emb + NorLIWC + Emotions + Intensity`: Achieved a Weighted F1 score of 0.812 (±0.018) and a ROC AUC score of 0.937 (±0.012) with LR.

Top Performing Configurations

- **Emb + Emotions:** With LR, this configuration yielded a Weighted F1 score of 0.819 (±0.020), a Micro F1 score of 0.819 (±0.019), and a ROC AUC score of 0.940 (±0.011).
- **Emb + Intensity:** Achieved similar performance with a Weighted F1 score of 0.818 (±0.021) and a ROC AUC score of 0.941 (±0.011) with LR.
- **Emb + Emotions + Intensity:** Also performed well, with a Weighted F1 score of 0.818 (±0.017) and a ROC AUC score of 0.940 (±0.011) with LR.
- **Emb + NorLIWC + Emotions:** Demonstrated strong performance with a Weighted F1 score of 0.813 (±0.020) and a ROC AUC score of 0.937 (±0.011) with LR.

Impact of Adding LIWC Features

- Adding LIWC features to the top-performing configurations (`Emb + LIWC + Emotions + Intensity`) resulted in a slight performance decrease. For example, the Weighted F1 score for this configuration was 0.807 (±0.022) with LR and 0.805 (±0.023) with XGB, slightly lower than the best-performing configurations without LIWC.

Summary
In general, LR models seem to perform slightly better than XGB models. The combination of sentence embeddings with other features, particularly emotions and intensity, consistently yielded the best results. Normalizing LIWC features also proved beneficial, leading to significant performance improvements when combined with sentence embeddings. Despite the slight performance decrease when adding LIWC to the top configurations, the integrated models still performed competitively.

The Confusion Matrix for one of the folds of the experiment `Emb + NorLIWC + Emotions + Intensity` with XGBoost is shown in Figure 6.1.1.

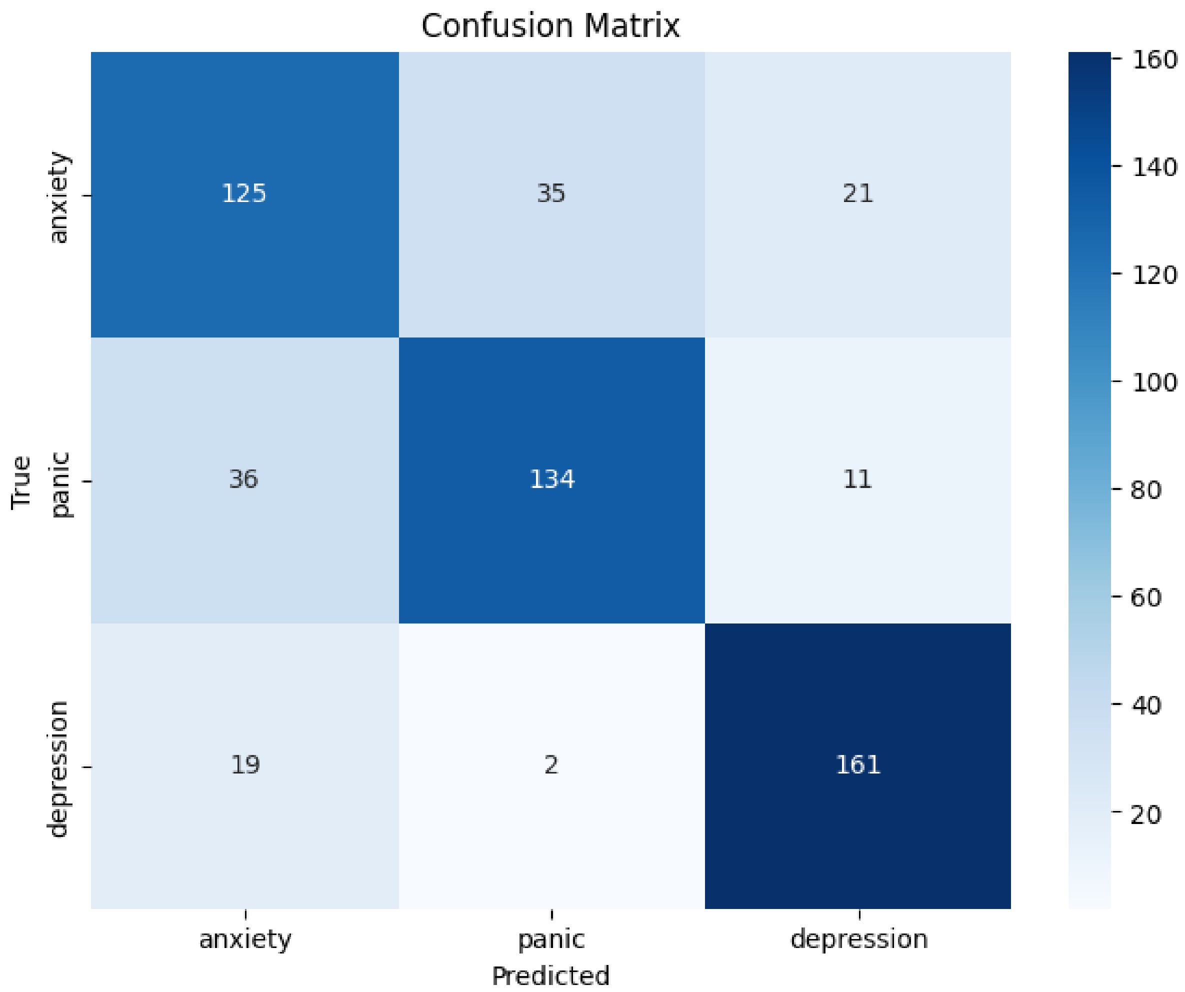


Figure 6.1.1: Confusion Matrix of the XGBoost model

6.1.3 Multi-layered Deep Learning Architecture Results

In this section, we present the results of our experiments, where we applied various combinations of LSTM, CNN, and Input (I) layers to our four features: sentence embeddings, emotions, intensity, and LIWC features. Each experiment involved different configurations of these layers, and the outcomes were evaluated using several evaluation metrics such as accuracy, losses, F1 score, and Matthews correlation coefficient (MCC). We used 10-fold cross-validation for all the experiments, evaluating the model on the test set; hence, each result for an evaluation metric is in the form (mean/std dev) of the 10 runs. The results are presented in Table 6.3.

In all the experiments, the layers follow the order "Sentence Embeddings, Emotions, Intensity, LIWC features." For instance, "LSTM (64) CNN (5,2) I I (32)" indicates:

- LSTM layer with 64 units applied to sentence embedding features.
- CNN layer with pool size=5 and strides=2 applied to emotions features.
- "I" denotes "Input," meaning that no extra layer is applied to the Intensity and LIWC features and that they were directly given as inputs to the neural network architecture.
- The last number (32) represents the batch size on which the model was trained.

Table 6.3: Results for Multi-layered Deep Learning Architecture

Experiment	Accuracy	Loss	F1	MCC
Input Input Input Input (32)	0.764/0.025	0.582/0.031	0.773/0.019	0.653/0.038
LSTM(64) CNN(5,2) I I (64)	0.785/0.029	0.517/0.049	0.790/0.023	0.681/0.043
LSTM(64) CNN(5,2) I I (64)	0.797/0.024	0.496/0.047	0.794/0.022	0.699/0.036
LSTM(32) CNN(5,2) I I (32)	0.791/0.024	0.513/0.048	0.797/0.022	0.689/0.037
LSTM(32) CNN(5,2) I I (32)	**0.807/0.018**	**0.487/0.048**	**0.807/0.014**	**0.711/0.027**
LSTM(128) CNN(5,2) I I (128)	0.792/0.027	0.499/0.051	0.794/0.025	0.691/0.040
LSTM(128) CNN(5,2) I I (128)	0.796/0.023	0.488/0.053	0.800/0.024	0.697/0.034
LSTM(32) CNN(5,2)+LSTM(32) I I (64)	**0.799/0.026**	**0.492/0.047**	**0.804/0.025**	**0.702/0.040**
LSTM(32) CNN(5,2) LSTM(32) I (64)	0.790/0.021	0.506/0.042	0.796/0.016	0.687/0.032
LSTM(32) CNN(3,1) I I (64)	**0.801/0.027**	**0.494/0.049**	**0.800/0.024**	**0.705/0.041**
LSTM(32) CNN(3,1) I I (32)	0.781/0.027	0.532/0.043	0.792/0.026	0.677/0.040

General Performance Overview:

1. **Direct inputs analysis:**

   - When all four features (sentence embeddings, emotions, intensity, LIWC) were directly input without applying LSTM or CNN layers, the model achieved an accuracy of 0.764 ($\pm$0.025), a loss of 0.582 ($\pm$0.031), an F1 score of 0.773 ($\pm$0.019), and an MCC of 0.653 ($\pm$0.038). This approach resulted in suboptimal performance compared to experiments where LSTM and CNN layers were strategically applied (below) to extract and process feature information.

2. **LSTM Units Analysis:**

   - **LSTM with 32 Units:** The results indicate that using LSTM with 32 units generally outperforms configurations with 64 and 128 units. For instance, the experiment with `LSTM(32) CNN (5,2) I I (32)` achieved an F1 score of 0.807 ($\pm$0.014), which is higher than any other LSTM configurations.
   - **LSTM with 64 Units:** The experiments with 64 units consistently showed lower performance compared to the 32 and 128-unit configurations, with the highest F1 score being 0.794 ($\pm$0.022).
   - **LSTM with 128 Units:** The performance of experiments using 128 units was moderately better than those with 64 units but generally inferior to the 32-unit configurations. The best performance in this category was 0.800 ($\pm$0.024) for F1 score, with a comparatively lower value of MCC with 0.697 ($\pm$0.034.)

3. **CNN Pool Size and Strides:**

   - The CNN layer with a pool size of 5 and strides of 2 (CNN (5,2)) was primarily used. The results showed significant improvements when combined with an LSTM layer, particularly in the `LSTM(32) CNN (5,2) I I (32)` configuration.
   - Alternative CNN configurations (CNN (3,1)) were also tested and showed promising results. For instance, the `LSTM(32) CNN (3,1) I I (64)` configuration achieved an F1 score of 0.800 ($\pm$0.024), with a high MCC of 0.705 ($\pm$0.041).

Experimentation with Additional Layers:

1. **Combination of CNN and LSTM Layers:**
   - `LSTM(32) CNN(5,2)+LSTM(32) I I (32)`: In this experiment, we applied a CNN layer followed by an LSTM layer to the emotions feature. This configuration performed exceptionally well, achieving an F1 score of 0.804 ($\pm$0.025).
   - `LSTM(32) CNN (5,2) LSTM(32) I (32)`: Here, we added an LSTM layer to the intensity feature as well. This experiment resulted in an F1 score of 0.796 ($\pm$0.016).

Top Performing Configurations:

1. **LSTM(32) CNN (5,2) I I (32):** This configuration yielded the best results, with an accuracy of 0.807 ($\pm$0.018), a loss of 0.487 ($\pm$0.048), an F1 score of 0.807 ($\pm$0.014), and an MCC of 0.711 ($\pm$0.027). This demonstrates that the combination of a 32-unit LSTM layer with a CNN (5,2) layer is highly effective.

2. **LSTM(32) CNN(5,2)+LSTM(32) I I (32):** This experiment also performed well, with an accuracy of 0.799 ($\pm$0.026), a loss of 0.492 ($\pm$0.047), an F1 score of 0.804 ($\pm$0.025), and an MCC of 0.702 ($\pm$0.040). This configuration shows the potential benefits of adding an additional LSTM layer on top of the CNN layer for the emotions feature.

3. **LSTM(32) CNN(3,1) I I (32):** This configuration achieved an accuracy of 0.801 ($\pm$0.027), a loss of 0.494 ($\pm$0.049), an F1 score of 0.800 ($\pm$0.024), and an MCC of 0.705 ($\pm$0.041), suggesting that alternative CNN configurations can also yield competitive results.

Summary:

The experiments demonstrate that using a 32-unit LSTM layer consistently yields better performance across various configurations. The combination of LSTM and CNN layers, particularly with a pool size of 5 and strides of 2, proved to be effective. Additionally, incorporating an extra LSTM layer on top of the CNN layer for the emotions feature showed significant improvement in performance. The top-performing configurations highlighted the importance of selecting appropriate layer combinations to enhance model performance.

The Confusion Matrix for one of the folds of the experiment `LSTM(32) CNN (5,2) I I (32)` is shown in Figure 6.1.2.

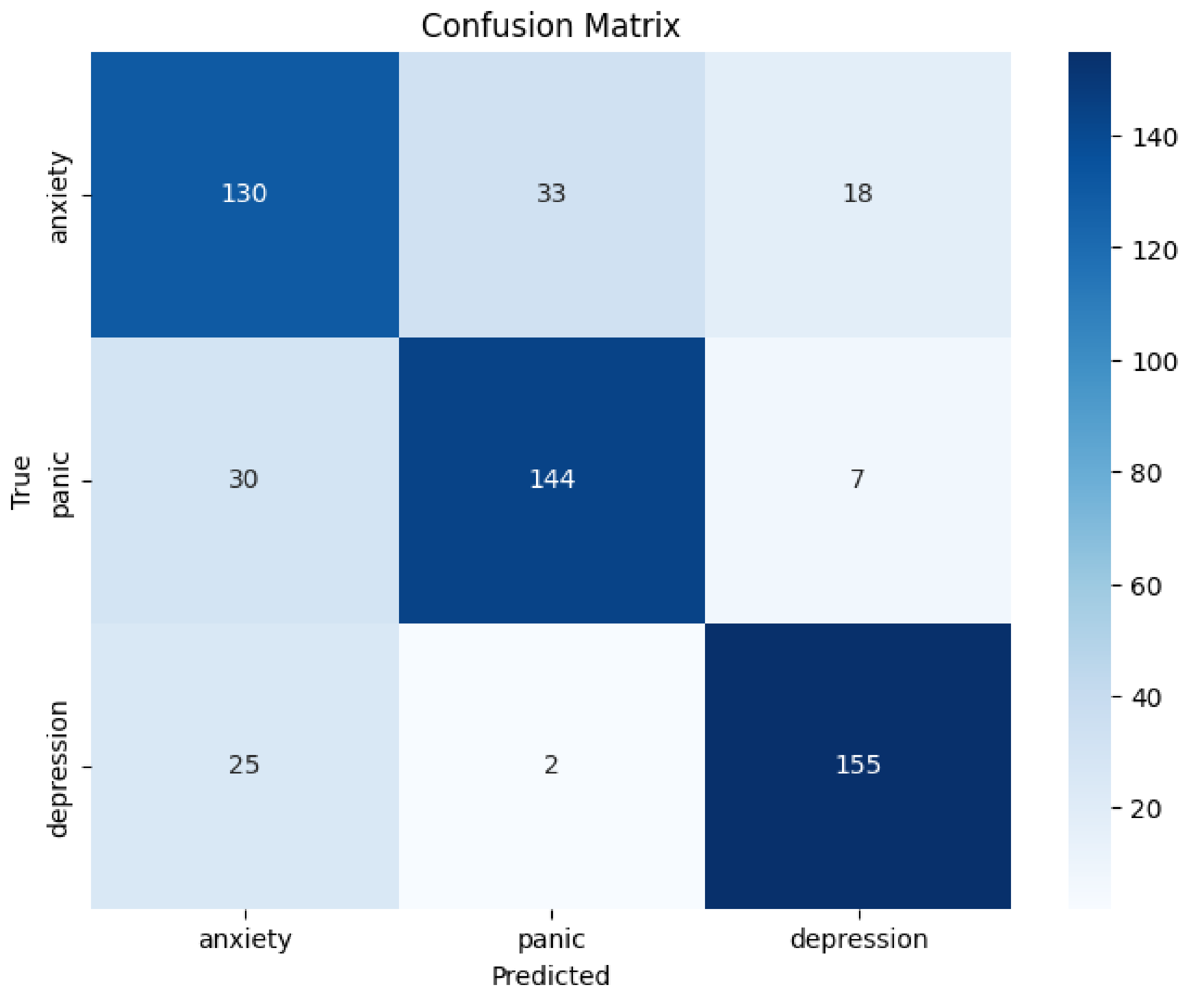


Figure 6.1.2: Confusion Matrix of the Deep Learning model

6.2 Discussions and Future Works

6.2.1 Discussions

The primary objective of this research was to evaluate the classifying capabilities of biomedical literature from social media. We developed and evaluated machine learning and deep learning models for the classification of anxiety, panic, and depression using data from Reddit. Various features were extracted to capture the semantic and emotional content of the posts, including contextual embeddings, LIWC features, emotion detection, and emotion intensity. Among the models tested, the Logistic Regression (LR) model achieved the highest F1 score of 81.9%, outperforming the Extreme Gradient Boosting (XGB) and multi-layered deep learning models, which achieved F1 scores of 80.3% and 80.7%, respectively.

These results indicate that the complexity of multiclass classification in mental health contexts is substantial. The underperformance of the XGB and deep learning models highlights the challenges in accurately modeling the nuanced expressions of mental health conditions in text data, which often include subtle semantic differences and emotional variations. The performance disparity between the models suggests that while traditional machine learning models like Logistic Regression can be effective, the inherent complexities of mental health classification necessitate more sophisticated approaches. The difficulty in distinguishing between closely related mental health conditions using text data underscores the need for further research and methodological improvements.

6.2.2 Future Works

Expanding the Dataset
The current study utilized a dataset of 5,441 posts, which, while sufficient for preliminary analysis, is relatively small for training complex models like XGB and deep learning networks. Future research should focus on collecting a larger dataset, encompassing more diverse and numerous posts for each mental health condition. A larger dataset can provide a more representative sample of the linguistic and emotional expressions associated with anxiety, panic, and depression, thereby enhancing the models' ability to generalize and improve overall performance.

Hyperparameter Tuning
This project applied LR and XGB models with default parameters. Future studies should involve systematic hyperparameter tuning to identify the optimal settings for each model, which can significantly improve their performance. Hyperparameter tuning for deep learning models, in particular, could involve experimenting with different network architectures, learning rates, batch sizes, and regularization techniques. Given the significant impact of hyperparameters on model performance, this approach is expected to yield better results and more robust models.

Leveraging Large Language Models (LLMs)
Advancements in NLP, particularly the development of Large Language Models (LLMs), offer promising new directions for this research. LLMs can be employed to extract detailed cause-and-effect relationships from the text, such as identifying specific triggers for anxiety or depression mentioned in social media posts. For instance, sentences like "I am anxious because I have an exam tomorrow" can be parsed to identify the cause ("exam tomorrow") and the effect ("anxious"). Utilizing LLMs to extract such relationships can enhance the interpretability of the models and provide deeper insights into the factors contributing to mental health conditions. Furthermore, this information can be used to construct knowledge graphs that map out the various causes and effects, offering a visual representation of the complex interplay between different factors. This approach can be particularly beneficial for the biomedical field, as it enables researchers and clinicians to understand better and address the multifaceted nature of mental health issues.

By addressing these areas, future research can significantly enhance the accuracy, robustness, and applicability of machine learning models in diagnosing mental health conditions based on social media data.

Bibliography

[1] Martin Prince, Vikram Patel, Shekhar Saxena, Mario Maj, Joanna Maselko, Michael R Phillips, and Atif Rahman. No health without mental health. *The lancet*, 370(9590):859–877, 2007.

[2] Rafael A Calvo, David N Milne, M Sazzad Hussain, and Helen Christensen. Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, 23(5):649–685, 2017.

[3] Esteban A Ríssola, Seyed Ali Bahrainian, and Fabio Crestani. A dataset for research on depression in social media. In *Proceedings of the 28th ACM conference on user modeling, adaptation and personalization*, pages 338–342, 2020.

[4] David William and Derwin Suhartono. Text-based depression detection on social media posts: A systematic literature review. *Procedia Computer Science*, 179:582–589, 2021.

[5] Elham Mohammadi, Hessam Amini, and Leila Kosseim. Quick and (maybe not so) easy detection of anorexia in social media posts. In *CLEF (Working Notes)*, 2019.

[6] Ankit Murarka, Balaji Radhakrishnan, and Sushma Ravichandran. Detection and classification of mental illnesses on social media using roberta. *arXiv preprint arXiv:2011.11226*, 2020.

[7] Yaakov Ophir, Refael Tikochinski, Christa SC Asterhan, Itay Sisso, and Roi Reichart. Deep neural networks detect suicide risk from textual facebook posts. *Scientific reports*, 10(1):16685, 2020.

[8] Bruce F Chorpita and David H Barlow. The development of anxiety: The role of control in the early environment. *The Neurotic Paradox, Vol 2*, pages 227–264, 2018.

[9] DSMTF American Psychiatric Association, American Psychiatric Association, et al. *Diagnostic and statistical manual of mental disorders: DSM-5*, volume 5. American psychiatric association Washington, DC, 2013.

[10] George A Clum, Gretchen A Clum, and Rebecca Surls. A meta-analysis of treatments for panic disorder. *Journal of consulting and clinical psychology*, 61(2):317, 1993.

[11] Argyris Stringaris. What is depression?, 2017.

[12] Sarah Edwards, Lucy Tinning, June SL Brown, Jed Boardman, and John Weinman. Reluctance to seek help and the perception of anxiety and depression in the united kingdom: A pilot vignette study. *The Journal of Nervous and Mental Disease*, 195(3):258–261, 2007.

[13] Sandra Mitrović, Oscar William Lithgow-Serrano, and Carlo Schillaci. Comparing panic and anxiety on a dataset collected from social media. In *Proceedings of the 9th Workshop on Computational Linguistics and Clinical Psychology (CLPsych 2024)*, pages 153–165, 2024.

[14] Sandra Mitrović, Fabio Frisone, Suryam Gupta, Chiara Lucifora, Dragana Čarapić, Carlo Schillaci, Samuele Di Giovanni, and Ayushi Singh. Annotating panic in social media using active learning, transformers and domain knowledge. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1269–1278. IEEE, 2023.

[15] Jina Kim, Jieon Lee, Eunil Park, and Jinyoung Han. A deep learning model for detecting mental illness from user content on social media. *Scientific reports*, 10(1):11846, 2020.

[16] Anamika Ahmed, Raihan Sultana, Md Tahmidur Rahman Ullas, Mariyam Begom, Md Muzahidul Islam Rahi, and Md Ashraful Alam. A machine learning approach to detect depression and anxiety using supervised learning. In *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pages 1–6. IEEE, 2020.

[17] Guangyao Shen, Jia Jia, Liqiang Nie, Fuli Feng, Cunjun Zhang, Tianrui Hu, Tat-Seng Chua, Wenwu Zhu, et al. Depression detection via harvesting social media: A multimodal dictionary learning solution. In *IJCAI*, pages 3838–3844, 2017.

[18] Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*, 2018.

[19] Waleed Ragheb, Jérôme Azé, Sandra Bringay, and Maximilien Servajean. Attention-based modeling for emotion detection and classification in textual conversations. *arXiv preprint arXiv:1906.07020*, 2019.

[20] Chenyang Huang, Amine Trabelsi, and Osmar R Zaïane. Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert. *arXiv preprint arXiv:1904.00132*, 2019.

[21] Raj Kumar Gupta and Yinping Yang. Crystalfeel at semeval-2018 task 1: Understanding and detecting emotion intensity using affective lexicons. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 256–263, 2018.

[22] Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. Meisd: A multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations. In *Proceedings of the 28th international conference on computational linguistics*, pages 4441–4453, 2020.

[23] Tomoyuki Kajiwara, Chenhui Chu, Noriko Takemura, Yuta Nakashima, and Hajime Nagahara. Wrime: A new dataset for emotional intensity estimation with subjective and objective annotations. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2095–2104, 2021.

[24] Laura Bostan, Evgeny Kim, and Roman Klinger. Goodnewseveryone: A corpus of news headlines annotated with emotions, semantic roles, and reader perception. *arXiv preprint arXiv:1912.03184*, 2019.

[25] Saif M Mohammad and Felipe Bravo-Marquez. Emotion intensities in tweets. *arXiv preprint arXiv:1708.03696*, 2017.

[26] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[27] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[28] Ryan L Boyd, Ashwini Ashokkumar, Sarah Seraj, and James W Pennebaker. The development and psychometric properties of liwc-22. *Austin, TX: University of Texas at Austin*, pages 1–47, 2022.

[29] Judy Hanwen Shen and Frank Rudzicz. Detecting anxiety through reddit. In *Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology—From Linguistic Signal to Clinical Reality*, pages 58–65, 2017.

[30] Marco Viviani, Cristina Crocamo, Matteo Mazzola, Francesco Bartoli, Giuseppe Carrà, and Gabriella Pasi. Assessing vulnerability to psychological distress during the covid-19 pandemic through the analysis of microblogging content. *Future Generation Computer Systems*, 125:446–459, 2021.

[31] Robinson Z Lumontod III. Seeing the invisible: Extracting signs of depression and suicidal ideation from college students' writing using liwc a computerized text analysis. *Int. J. Res. Stud. Educ*, 9(4):31–44, 2020.

[32] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[33] Saif M. Mohammad. Word affect intensities. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan, 2018.

[34] Giorgio Leonardi, Stefania Montani, and Manuel Striani. Novel deep learning architectures for haemodialysis time series classification. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 26(2):91–99, 2022.

DECLARATION

I, **Suryam Gupta** (Reg. No.: **I19MA038**), Department of **Mathematics** have completed my Integrated M.Sc. work with following details:

Title of Thesis: "**Evaluating Psychometric Features and Contextual Embeddings Potential for Mental Disorder Classification in Short Biomedical Texts**"

Supervisor(s):

1. **Dr. Amit Sharma**, Assistant Professor, Department of Mathematics, SVNIT, Surat
2. **Dr. Sandra Mitrović**, Full-Time Teacher and Postdoctoral Researcher, IDSIA, USI/SUPSI, Switzerland

I declare that the said work is result of my research activity created by me during my Integrated M.Sc. program, and proper references have been cited as applicable. The thesis does not involve any plagiarised contents as enumerated in **Policy on Prevention of Plagiarism in M.Tech. and PhD Dissertations being submitted at SVNIT Surat.**' The thesis has been scanned through **Turnitin** antiplagiarism software. I also undertake that, in case any plagiarised content is reported in my Thesis being submitted here with, the action be initiated as per the policy of the Institute on antiplagiarism.

Suryam Gupta

Reg. No.: I19MA038 Mail Id: suryamgupta12345@gmail.com

Countersignature(s)

Supervisor(s):

1.

Forwarded by

Head of the Department

Department of Mathematics Signature

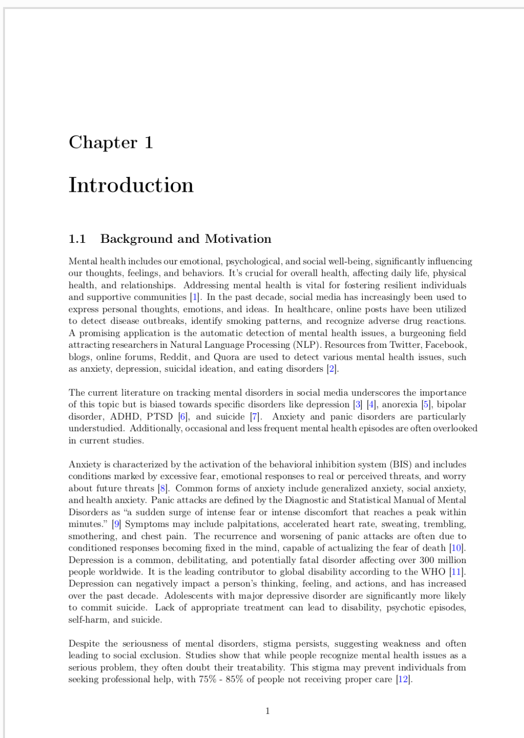
To,
The Dean (Academics)
SVNIT Surat
Surat 395007

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Suryam Gupta
Assignment title:	Quick Submit
Submission title:	Dissertation
File name:	Final_draft_for_plagiarism_check_1.pdf
File size:	2.55M
Page count:	54
Word count:	16,410
Character count:	89,026
Submission date:	21-Jun-2024 01:03PM (UTC+0530)
Submission ID:	2406178724


Chapter 1

Introduction

1.1 Background and Motivation

Mental health includes our emotional, psychological, and social well-being, significantly influencing our thoughts, feelings, and behaviors. It's crucial for overall health, affecting daily life, physical health, and relationships. Addressing mental health is vital for fostering resilient individuals and supportive communities [1]. In the past decade, social media has increasingly been used to express personal thoughts, emotions, and ideas. In healthcare, online posts have been utilized to detect disease outbreaks, identify smoking patterns, and recognize adverse drug reactions. A promising application is the automatic detection of mental health issues, a burgeoning field attracting researchers in Natural Language Processing (NLP). Resources from Twitter, Facebook, blogs, online forums, Reddit, and Quora are used to detect various mental health issues, such as anxiety, depression, suicidal ideation, and eating disorders [2].

The current literature on tracking mental disorders in social media underscores the importance of this topic but is biased towards specific disorders like depression [3] [4], anorexia [5], bipolar disorder, ADHD, PTSD [6], and suicide [7]. Anxiety and panic disorders are particularly understudied. Additionally, occasional and less frequent mental health episodes are often overlooked in current studies.

Anxiety is characterized by the activation of the behavioral inhibition system (BIS) and includes conditions marked by excessive fear, emotional responses to real or perceived threats, and worry about future threats [8]. Common forms of anxiety include generalized anxiety, social anxiety, and health anxiety. Panic attacks are defined by the Diagnostic and Statistical Manual of Mental Disorders as "a sudden surge of intense fear or intense discomfort that reaches a peak within minutes." [9] Symptoms may include palpitations, accelerated heart rate, sweating, trembling, smothering, and chest pain. The recurrence and worsening of panic attacks are often due to conditioned responses becoming fixed in the mind, capable of actualizing the fear of death [10]. Depression is a common, debilitating, and potentially fatal disorder affecting over 300 million people worldwide. It is the leading contributor to global disability according to the WHO [11]. Depression can negatively impact a person's thinking, feeling, and actions, and has increased over the past decade. Adolescents with major depressive disorder are significantly more likely to commit suicide. Lack of appropriate treatment can lead to disability, psychotic episodes, self-harm, and suicide.

Despite the seriousness of mental disorders, stigma persists, suggesting weakness and often leading to social exclusion. Studies show that while people recognize mental health issues as a serious problem, they often doubt their treatability. This stigma may prevent individuals from seeking professional help, with 75% - 85% of people not receiving proper care [12].

1

Dissertation

ORIGINALITY REPORT

19%	14%	12%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liverpool John Moores University Student Paper	1%
2	arxiv.org Internet Source	1%
3	saifmohammad.com Internet Source	<1%
4	www.liwc.app Internet Source	<1%
5	mars.gmu.edu Internet Source	<1%
6	consortiacademia.org Internet Source	<1%
7	Submitted to University of Huddersfield Student Paper	<1%
8	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1%

9	Submitted to Higher Education Commission Pakistan Student Paper	<1%
10	Sandra Mitrović, Fabio Frisone, Suryam Gupta, Chiara Lucifora et al. "Annotating Panic in Social Media using Active Learning, Transformers and Domain Knowledge", 2023 IEEE International Conference on Data Mining Workshops (ICDMW), 2023 Publication	<1%
11	www.saifmohammad.com Internet Source	<1%
12	Submitted to Loughborough University Student Paper	<1%
13	Submitted to University of Exeter Student Paper	<1%
14	supp.apa.org Internet Source	<1%
15	www.arxiv-vanity.com Internet Source	<1%
16	infoscience.epfl.ch Internet Source	<1%
17	Submitted to Suleyman Demirel University, Kazakhstan Student Paper	<1%

18	Jun Zhang, Yanrong Guo. "Multilevel depression status detection based on fine-grained prompt learning", Pattern Recognition Letters, 2024 Publication	<1 %
19	Submitted to Silicon Institute of Technology Student Paper	<1 %
20	Akshay Kulkarni, Adarsha Shivananda, Anoosh Kulkarni, Dilip Gudivada. "Applied Generative AI for Beginners", Springer Science and Business Media LLC, 2023 Publication	<1 %
21	Li Ruan, Yu Bai, Shaoning Li, Jiaxun Lv, Tianyuan Zhang, Limin Xiao, Haiguang Fang, Chunhao Wang, Yunzhi Xue. "Cloud Workload Turning Points Prediction via Cloud Feature-Enhanced Deep Learning", IEEE Transactions on Cloud Computing, 2023 Publication	<1 %
22	www.mdpi.com Internet Source	<1 %
23	Submitted to St George's Hospital Medical School Student Paper	<1 %
24	www.researchsquare.com Internet Source	<1 %

25	Piotr Myśliwiec, Andrzej Kubit, Paulina Szawara. "Optimization of 2024-T3 Aluminum Alloy Friction Stir Welding Using Random Forest, XGBoost, and MLP Machine Learning Techniques", Materials, 2024 Publication	<1 %
26	Submitted to University of Durham Student Paper	<1 %
27	fastercapital.com Internet Source	<1 %
28	huggingface.co Internet Source	<1 %
29	spectrum.library.concordia.ca Internet Source	<1 %
30	Submitted to University of Hertfordshire Student Paper	<1 %
31	www.geeksforgeeks.org Internet Source	<1 %
32	"Early Detection of Mental Health Disorders by Social Media Monitoring", Springer Science and Business Media LLC, 2022 Publication	<1 %
33	docs.google.com Internet Source	<1 %
34	mafiadoc.com Internet Source	

<1 %

35	www.capitalone.com Internet Source	<1 %
36	www.theseus.fi Internet Source	<1 %
37	Submitted to National University of Ireland, Galway Student Paper	<1 %
38	Submitted to Associatie K.U.Leuven Student Paper	<1 %
39	Submitted to Icon College of Technology and Management Student Paper	<1 %
40	Submitted to Westcliff University Student Paper	<1 %
41	David D'Acunto, Annamaria Tuan, Daniele Dalli. "chapter 12 Are Online Reviews Helpful for Consumers?", IGI Global, 2020 Publication	<1 %
42	Parag Verma, Shayan Anwar, Shadab Khan, Sunil B Mane. "Network Intrusion Detection Using Clustering and Gradient Boosting", 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018	<1 %

Publication

43	Submitted to University of Mauritius Student Paper	<1 %
44	Submitted to Unizin, LLC Student Paper	<1 %
45	assets.researchsquare.com Internet Source	<1 %
46	aclanthology.lst.uni-saarland.de Internet Source	<1 %
47	Submitted to islamicuniversity Student Paper	<1 %
48	openaccess.hacettepe.edu.tr Internet Source	<1 %
49	www.pythonicly.com Internet Source	<1 %
50	Submitted to Michigan Technological University Student Paper	<1 %
51	Nourah Alswaidan, Mohamed El Bachir Menai. "A survey of state-of-the-art approaches for emotion recognition in text", Knowledge and Information Systems, 2020 Publication	<1 %
52	Submitted to Sai University Student Paper	<1 %

53	Submitted to University of Alabama at Birmingham Student Paper	<1 %
54	www.tritonia.fi Internet Source	<1 %
55	Submitted to Saint Francis College Student Paper	<1 %
56	Submitted to University of Central Florida Student Paper	<1 %
57	Submitted to University of Oulu Student Paper	<1 %
58	aclanthology.org Internet Source	<1 %
59	dokumen.pub Internet Source	<1 %
60	ipfs.io Internet Source	<1 %
61	mssc.mu.edu Internet Source	<1 %
62	Submitted to University of Bristol Student Paper	<1 %
63	Submitted to University of Lancaster Student Paper	<1 %

64	Valerio Basile, Francesco Cauteruccio, Giorgio Terracina. "How Dramatic Events Can Affect Emotionality in Social Posting: The Impact of COVID-19 on Reddit", Future Internet, 2021 Publication	<1%
65	fictionbook.ru Internet Source	<1%
66	theses.liacs.nl Internet Source	<1%
67	"Advances in Data Science and Information Engineering", Springer Science and Business Media LLC, 2021 Publication	<1%
68	Lecture Notes in Computer Science, 2008. Publication	<1%
69	Submitted to Queen's University of Belfast Student Paper	<1%
70	Sahil Negi, Sudipta Kumar Das, Rigzen Bodh. "Credit Card Fraud Detection using Deep and Machine Learning", 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022 Publication	<1%
71	Submitted to Sheffield Hallam University Student Paper	<1%

72 Soumitra Ghosh, Asif Ekbal, Pushpak Bhattacharyya. "VAD-assisted multitask transformer framework for emotion recognition and intensity prediction on suicide notes", Information Processing & Management, 2023
Publication
<1 %

73 doi.org
Internet Source
<1 %

74 open.metu.edu.tr
Internet Source
<1 %

75 opus.lib.uts.edu.au
Internet Source
<1 %

76 "Artificial Intelligence in Medicine", Springer Science and Business Media LLC, 2020
Publication
<1 %

77 David Paper. "TensorFlow 2.x in the Colaboratory Cloud", Springer Science and Business Media LLC, 2021
Publication
<1 %

78 Submitted to Southampton Solent University
Student Paper
<1 %

79 Submitted to The Hong Kong Polytechnic University
Student Paper
<1 %

80 dustintench.pbworks.com

	Internet Source	<1%
81	www.frontiersin.org Internet Source	<1%
82	Submitted to City University of Hong Kong Student Paper	<1%
83	Kaixin Yang, Long Liu, Yalu Wen. "The impact of Bayesian optimization on feature selection", Scientific Reports, 2024 Publication	<1%
84	Submitted to Universidad Carlos III de Madrid - EUR Student Paper	<1%
85	Yung-Chun Chang, Yan-Chun Hsing. "Emotion-infused deep neural network for emotionally resonant conversation", Applied Soft Computing, 2021 Publication	<1%
86	link.springer.com Internet Source	<1%
87	Submitted to City University Student Paper	<1%
88	Submitted to Coventry University Student Paper	<1%
89	Submitted to Deakin University Student Paper	<1%

90	Giovanna Maria Dora Dore, Arya D. McCarthy, James A. Scharf. "A Free Press, If You Can Keep It", Springer Science and Business Media LLC, 2023 Publication	<1 %
91	Submitted to Middle East College of Information Technology Student Paper	<1 %
92	Nguyen Nguyen, Joseph Johnson, Michael Tsiros. "Unlimited Testing: Let's Test Your Emails with AI", Marketing Science, 2024 Publication	<1 %
93	Submitted to University of Stellenbosch, South Africa Student Paper	<1 %
94	docs.semanticbrandscore.com Internet Source	<1 %
95	www.cmi.ac.in Internet Source	<1 %
96	Michael M. Tadesse, Hongfei Lin, Bo Xu, Liang Yang. "Personality Predictions Based on User Behaviour on the Facebook Social Media Platform", IEEE Access, 2018 Publication	<1 %
97	Submitted to The Robert Gordon University Student Paper	<1 %

98	Submitted to The University of Manchester Student Paper	<1 %
99	Submitted to Tilburg University Student Paper	<1 %
100	Submitted to University of Edinburgh Student Paper	<1 %
101	Submitted to University of the Philippines Los Banos Student Paper	<1 %
102	Submitted to ebsu Student Paper	<1 %
103	www.photonics.com Internet Source	<1 %
104	Submitted to Berner Fachhochschule Student Paper	<1 %
105	Ross E. Heath, James M. Gelb, George L. Tipple. "Classification of Clutter Types in Active Sonar Using Spatial Image Processing Techniques", OCEANS 2007 - Europe, 2007 Publication	<1 %
106	Sancheng Peng, Lihong Cao, Yongmei Zhou, Zhouhao Ouyang, Aimin Yang, Xinguang Li, Weijia Jia, Shui Yu. "A survey on deep learning for textual emotion analysis in social networks", Digital Communications and Networks, 2021	<1 %

Publication

107	Submitted to University of College Cork Student Paper	<1%
108	Submitted to University of Hull Student Paper	<1%
109	Submitted to University of Sheffield Student Paper	<1%
110	core.ac.uk Internet Source	<1%
111	dr.ntu.edu.sg Internet Source	<1%
112	medium.com Internet Source	<1%
113	www.irjmets.com Internet Source	<1%
114	Submitted to Manchester Metropolitan University Student Paper	<1%
115	Van-Nguyen Pham, Duc-Tai Le, Junghyun Bum, Seong Ho Kim, Su Jeong Song, Hyunseung Choo. "Discriminative-Region Multi-Label Classification of Ultra-Widefield Fundus Images", Bioengineering, 2023 Publication	<1%
116	anikina.de Internet Source	

		<1 %
117	github.com Internet Source	<1 %
118	rocknyc.live Internet Source	<1 %
119	Ashritha R Murthy, K M Anil Kumar. "A Review of Different Approaches for Detecting Emotion from Text", IOP Conference Series: Materials Science and Engineering, 2021 Publication	<1 %
120	Submitted to Hong Kong Baptist University Student Paper	<1 %
121	Submitted to Indian School of Business Student Paper	<1 %
122	archive.saulibrary.edu.bd:8080 Internet Source	<1 %
123	dspace.lib.uom.gr Internet Source	<1 %
124	themighty.com Internet Source	<1 %
125	www.dangersemo.com Internet Source	<1 %
126	www.fastercapital.com Internet Source	<1 %