PSYCHOLOGICAL DISORDER ASSESSMENT USING MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING

Project report submitted
in partial fulfillment of the requirement for the degree of
Bachelor of Engineering in Information Technology

BY

Rajdeep Mallick (002011001050)
Arup Kumar Roy (002011001101)
Souvik Naskar (002011001105)

Under the guidance of
Dr. Sruti Gan Chaudhuri

Department of Information Technology,
Faculty of Engineering and Technology,
Jadavpur University, Salt Lake Campus
2023-2024

Department of Information Technology

Faculty of Engineering and Technology

Jadavpur University

BONAFIDE CERTIFICATE

This is to certify that this project report entitled "**PSYCHOLOGICAL DISORDER ASSESSMENT USING MACHING LEARNING AND NATURAL LANGUAGE PROCESSING**" submitted to **Department of Information Technology, Jadavpur University, Salt Lake Campus, Kolkata**, is a bonafide record of work done by **Rajdeep Mallick (Registration No: 153777 of 2020-21), Arup Kumar Roy (Registration No: 153804 of 2020-21) and Souvik Naskar (Registration No: 153808 of 2020-21)** under my supervision from **04.07.2023** to **15.05.2024**.

Dr. Sruti Gan Chaudhuri
Assistant Professor

Countersigned By:
Bibhas Chandra Dhara,
Head Of the Department,
Department of Information Technology

Declaration by Students

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we shall take full responsibility for it. All information, materials, and methods that are not original to this work have been properly referenced and cited. We also declare that no part of this project work has been submitted for the award of any other degree prior to this date.

Rajdeep Mallick
Roll No.- 002011001050

Arup Kumar Roy
Roll No.- 002011001101

Souvik Naskar
Roll No.- 002011001105

Place: Kolkata
Date: 22/05/2024

Acknowledgments

We would like to acknowledge those without those help and guidance this project would not have been possible. These people have always been a tremendous support whenever we needed some motivation and guidance while working on this project.

First, we would like to thank our supervisor Dr. Sruti Gan Chaudhuri who presented us with the opportunity to learn and state our findings through this project. Without her guidance and support our project would not have been possible.

Secondly, we would like to extend our gratitude to the University and the Department of Information Technology for providing us with the resources to make this project a success. The technical staff of the software labs kept the systems running and in turn helped us to work on the project without and worries.

Finally, we would like to thank our parents and family for providing us with an environment conducive to learning and their affection and care which was a constant source of inspiration for us.

Department of Information Technology
Jadavpur University
Salt Lake Campus, Kolkata

Rajdeep Mallick
Roll No.- 002011001050

Arup Kumar Roy
Roll no: 002011001101

Souvik Naskar
Roll no: 002011001105

JADAVPUR UNIVERSITY
Dept. of Information Technology

Vision:

To provide young undergraduate and postgraduate students a responsive research environment and quality education in Information Technology to contribute in education, industry and society at large.

Mission:

M1: To nurture and strengthen professional potential of undergraduate and postgraduate students to the highest level.

M2: To provide international standard infrastructure for quality teaching, research and development in Information Technology.

M3: To undertake research challenges to explore new vistas of Information and Communication Technology for sustainable development in a value-based society.

M4: To encourage teamwork for undertaking real life and global challenges.

Program Educational Objectives (PEOs):

Graduates should be able to:

PEO1: Demonstrate recognizable expertise to solve problems in the analysis, design, implementation and evaluation of smart, distributed, and secured software systems.

PEO2: Exhibit sustained learning capability and ability to adapt to a constantly changing field of Information Technology through professional development, and self-learning.

PEO3: To undertake research challenges to explore new vistas of Information and Communication Technology for sustainable development in a value-based society.

PEO4: Show leadership qualities and initiative to ethically advance professional and organizational goals through collaboration with others of diverse interdisciplinary backgrounds.

Mission - PEO matrix:

Ms/ PEOs	**M1**	**M2**	**M3**	**M4**
PEO1	3	2	2	1
PEO2	2	3	2	1
PEO3	2	2	3	1
PEO4	1	2	2	3

(3 – Strong, 2 – Moderate and 1 – Weak)

Program Specific Outcomes (PSOs):

At the end of the program a student will be able to:

PSO1: Apply the principles of theoretical and practical aspects of ever evolving Programming & Software Technology in solving real life problems efficiently.

PSO2: Develop secured software systems considering constantly changing paradigms of communication and computation of web enabled distributed Systems.

PSO3: Design ethical solutions of global challenges by applying intelligent data science & management techniques on suitable modern computational platforms through interdisciplinary collaboration.

Abstract

The development of an explainable intelligence-driven balanced decision tree approach for psychological disorder assessment, and the use of natural language processing (NLP) techniques to analyze stress-related discourse from online platforms like Reddit.

The first part of the project explores the use of decision tree models enhanced with explainable intelligence to improve the accuracy and interpretability of psychological disorder assessment. This approach involves transforming and encoding raw data, segregating and labeling attributes, prioritizing queries for elimination, training and testing the model, and providing explanations for predictions.

The second part utilizes the Dreaddit dataset, which comprises Reddit posts labeled as “Stress” or “Not Stress.” NLP techniques such as sentiment analysis, topic modeling, and word embeddings are employed to analyze stress-related discourse and identify patterns and trends. Various machine learning (ML) models are evaluated, including traditional methods like Support Vector Machines (SVMs), logistic regression, Naive Bayes, and Random Forest, as well as neural network approaches like Artificial Neural Networks (ANNs), LSTM, Bi-LSTM, and GRU.

The project aims to contribute to the field of psychological disorder assessment by exploring the potential of both decision tree modeling and NLP analysis. While the decision tree approach faced limitations, the NLP analysis yielded promising results. The combination of Fasttext embeddings and ANNs proved particularly effective in identifying signs of stress in Reddit posts. The project highlights the potential of using social media data to gain insights into mental health challenges and develop more effective interventions.

TABLE OF CONTENTS

1. Introduction **1**

2. Related Works **3**

3. Part One Model & Work **4**

3.1 Model 4

3.2 Transform Encode Phase 4

3.3 Segregate Label Phase 6

3.4 Q-Prioritization Phase 6

3.5 Train-Test Phase 8

3.5.1 Proposed Balanced Decision Tree Method 10

3.6 Prediction Interpret Phase 11

3.6.1 Permuted Feature Importance Method 12

3.6.2 Contrastive Explanation Method 13

3.7 Our Implementation & Findings 13

3.7.1 Dataset Overview 13

3.7.2 Data Preprocessing 14

3.7.3 Feature Selection 14

3.7.4 Model Selection and Training-Testing 16

3.7.5 Interpretation Phase 17

3.8 Limitations 18

4. Part Two Work **20**

4.1 Dreaddit Dataset Overview 20

4.2 NLP Pipeline 21

4.3 Dataset Collection 22

4.4 Post Body (Text) 23

4.5 Text Preprocessing 23

4.6 Word Embedding 24

4.7 Model Experimentation 26

4.7.1 Models Used in Our Work 28

4.8 Results & Findings 30

4.8.1 Only Text & Any Confidence 30

4.8.2 Only Text & 80% Confidence 31

4.8.3 Word Embedding + Features & Any Confidence 32

4.8.4 Word Embedding + Features & 80% Confidence 33

4.9 Testing with Demo Data 35

4.10 Limitations 36

5. Conclusion **37**

6. Future Work **38**

List of Figures

1. Proposed Balanced decision tree approach 11
2. Pertinent positive sample example 13
3. Feature Selection for Anxiety Data 15
4. Classification Reports for Decision Tree Model 17
5. Decision Tree Model (Anxiety Data) 18
6. Extracted features from dataset 21
7. Implemented NLP Pipeline 22
8. Original Research vs Our Work word embeddings 26
9. Model Experimentation Setup 27
10. Word Embeddings with Any confidence 30
11. Word embeddings with 80% confidence 31
12. Word embeddings with features and any confidence 32
13. Word embeddings with features and 80% confidence 33

List of Tables

1. A sample weight score computation illustration 5
2. DASS 42 score table for Depression, Anxiety, Stress 5
3. Text Preprocessing Example 24
4. Comparative study table of original vs our work results 34
5. Test with Demo data table 35

List of Algorithms

1. Pseudocode for Q-prioritization phase 8
2. Decision tree building from training instances of data partition *P* 9
3. Pseudocode for Information gain 10
4. Pseudocode for permuted feature importance method 12

1. Introduction

In recent years, the global landscape of mental health has become increasingly prominent, with the World Health Organization (WHO) emphasizing the significance of addressing psychological disorders such as anxiety, stress, and depression. The prevalence of these conditions is on the rise, fueled by personal and professional instabilities, societal pressures, and various other factors. As individuals navigate through the complexities of modern life, the need for accurate assessment and effective treatment of psychological disorders has become more pressing than ever before.

However, the task of assessing and treating psychological disorders presents a multifaceted challenge. One of the primary obstacles lies in the overlapping symptoms exhibited by different disorders, making accurate diagnosis and classification a daunting endeavor. Traditional diagnostic methods often rely heavily on subjective interpretations and are prone to human error, leading to misdiagnosis and inadequate treatment. In this context, there is a growing recognition of the potential of artificial intelligence (AI) and machine learning (ML) to revolutionize the field of mental health assessment.

One approach that holds promise in this domain is the utilization of decision tree models driven by explainable intelligence. These models offer a structured framework for analyzing complex datasets and making informed decisions based on a series of logical rules. By leveraging explainable intelligence, these decision tree models not only provide accurate predictions but also offer insights into the underlying reasoning process, thereby enhancing transparency and interpretability.

The aim of this project is to delve into the intricacies of an Explainable Intelligence Driven Balanced Decision Tree Approach for Psychological Disorder Assessment. This approach represents a fusion of advanced machine learning techniques with principles of explainable AI, offering a novel paradigm for enhancing the accuracy and interpretability of psychological disorder assessment. By developing a comprehensive understanding of this approach, we aim to contribute to the ongoing efforts to revolutionize mental health assessment and treatment.

Throughout this study, we will explore the theoretical foundations of decision tree algorithms, delve into the concept of explainable intelligence, and examine the practical implications of applying this approach to psychological disorder assessment. Through empirical analysis and case studies, we seek to demonstrate the efficacy and potential of our proposed methodology in addressing the challenges inherent in mental health diagnosis and treatment.

In addition to exploring the Explainable Intelligence Driven Balanced Decision Tree Approach, our second part of the project incorporates Natural Language Processing (NLP) techniques to analyze stress-related discourse from online platforms such as Reddit. By leveraging the vast amount of textual data available on these platforms, we aim to gain deeper insights into the experiences and expressions of individuals dealing with stress. Through sentiment analysis, topic modeling, and other NLP methods, we seek to uncover patterns, trends, and underlying factors contributing to stress manifestation.

Various machine learning (ML) models such as random forests, neural networks, decision trees, and CNNs have been applied to predict and diagnose psychological disorders. However, achieving diagnostic accuracy poses a challenge due to the heterogeneous nature of these disorders. To address this, we have conducted comparative analyses among these algorithms, utilizing diverse datasets from platforms to establish psychological assessment tools such as DASS 21 and PHQ-9. This comparative approach aims to identify the strengths and limitations of each algorithm, guiding the selection of optimal models for specific diagnostic tasks.

Through empirical validation and case studies, we endeavor to demonstrate the efficacy and practical utility of our approach in real-world settings. By shedding light on the nuanced interplay between individual characteristics, environmental factors, and psychological well-being, we aspire to inform more targeted and personalized interventions for stress management and mental health support.

In summary, our project represents a concerted effort to advance the field of psychological disorder assessment through a multidimensional approach. By combining the power of decision tree modeling with the insights gleaned from NLP analysis, we aim to develop a nuanced understanding of stress and other psychological disorders, paving the way for more effective interventions and improved outcomes for individuals experiencing mental health challenges.

2. Related Works

The Dreaddit dataset project builds upon a substantial body of prior research in the field of stress detection and analysis in social media. Alarcao and Fonseca [1] highlighted the potential of using various signals, including social media data, for stress detection, utilizing machine learning techniques to analyze different data types. Similarly, Anjume et al. [2] explored the application of machine learning models to identify signs of mental health disorders in diverse datasets, including social media posts. Dabek and Caban [3] developed models to assess psychological conditions using neural networks, demonstrating their effectiveness in health prediction. Muhammad Abdul-Mageed and Ungar [4] employed deep learning approaches to predict fine-grained emotions, illustrating the potential of advanced neural networks in mental health detection. De Choudhury et al. [5] emphasized the importance of contextual information and temporal patterns in detecting psychological stress. Devlin et al. [6] demonstrated the applicability of advanced transformer models to informal social media texts, which require robust models to accurately detect emotional content. Kim [7] applied convolutional neural networks for emotion detection, supporting the Dreaddit project's methodologies. Guntuku et al. [8] developed models to assess stress levels from linguistic cues in social media text, demonstrating the effectiveness of sentiment analysis and linguistic feature extraction. Mikolov et al. [9] further explored the use of word embeddings in emotion detection tasks. Winata et al. [10] demonstrated the applicability of advanced neural network techniques to detect psychological stress from spoken language, further supporting the methodologies used in the Dreaddit project.

3. Part One Model & Work

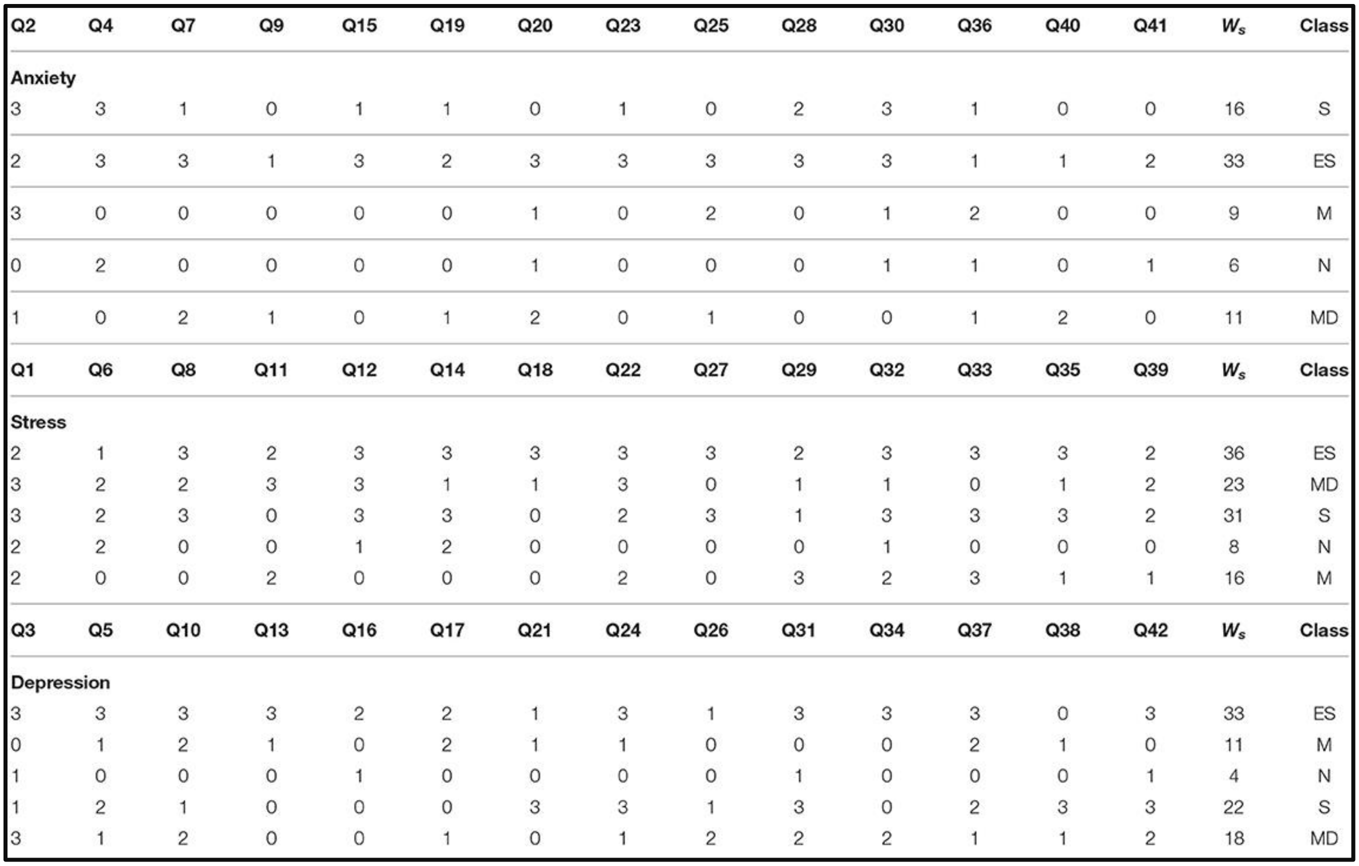
3.1 Model

In our research, we focus on employing an explainable intelligence-based predictive model to identify physiological disorders such as anxiety, stress, and depression. This model operates through a series of distinct phases. Initially, raw data is transformed and encoded into an interpretable template. Following this, data attributes are segregated and labeled according to the respective disorders they represent. Subsequently, less significant queries or attributes are prioritized for elimination. The model then undergoes training and testing using a machine intelligence algorithm to learn from examples and make predictions. Post-prediction, explanations for the outcomes are provided to enhance interpretability. Finally, a comprehensive validation process ensures the accuracy and reliability of the model's predictions. These phases collectively form a structured framework for the recognition and understanding of physiological disorders, laying the foundation for further analysis and exploration.

3.2 Transform Encode Phase

In the Transform-Encode phase, the initial step involves gathering retrieved data samples. These raw, unprocessed data require structuring into a suitable template for easy interpretation. After the collection, once the data samples are loaded into the Python platform with appropriate libraries, the next crucial step involves data preprocessing. Detection of missing values is essential as they can significantly impact data processing and analysis. Any entries that fall outside their predefined range are flagged as missing values. These values are then counted, identified, and replaced with the most common data value for that specific column to ensure data integrity consistency.

Following the handling of missing values, data encoding is performed to facilitate analysis. Based on severity levels, the responses are assigned numerical values typically ranging from 1 to 4. Additionally, a rating point is determined on a scale ranging from 0 to 3 by subtracting 1 from the encoded value. This systematic encoding process aids in standardizing the data and simplifying subsequent analysis and interpretation, setting the stage for further exploration and model development.



Q2	Q4	Q7	Q9	Q15	Q19	Q20	Q23	Q25	Q28	Q30	Q36	Q40	Q41	W_s	Class
Anxiety															
3	3	1	0	1	1	0	1	0	2	3	1	0	0	16	S
2	3	3	1	3	2	3	3	3	3	3	1	1	2	33	ES
3	0	0	0	0	0	1	0	2	0	1	2	0	0	9	M
0	2	0	0	0	0	1	0	0	0	1	1	0	1	6	N
1	0	2	1	0	1	2	0	1	0	0	1	2	0	11	MD
Q1	Q6	Q8	Q11	Q12	Q14	Q18	Q22	Q27	Q29	Q32	Q33	Q35	Q39	W_s	Class
Stress															
2	1	3	2	3	3	3	3	3	2	3	3	3	2	36	ES
3	2	2	3	3	1	1	3	0	1	1	0	1	2	23	MD
3	2	3	0	3	3	0	2	3	1	3	3	3	2	31	S
2	2	0	0	1	2	0	0	0	0	1	0	0	0	8	N
2	0	0	2	0	0	0	2	0	3	2	3	1	1	16	M
Q3	Q5	Q10	Q13	Q16	Q17	Q21	Q24	Q26	Q31	Q34	Q37	Q38	Q42	W_s	Class
Depression															
3	3	3	3	2	2	1	3	1	3	3	3	0	3	33	ES
0	1	2	1	0	2	1	1	0	0	0	2	1	0	11	M
1	0	0	0	1	0	0	0	0	1	0	0	0	1	4	N
1	2	1	0	0	0	3	3	1	3	0	2	3	3	22	S
3	1	2	0	0	1	0	1	2	2	2	1	1	2	18	MD

Table 1: A sample weight score computation illustration

DASS (42) Scoring	**Depression**	**Anxiety**	**Stress**
Normal	0-9	0-7	0-14
Mild	10-13	08-09	15-18
Moderate	14-20	10-14	19-25
Severe	21-27	15-19	26-33
Extremely Severe	28+	20+	34+

Table 2: DASS 42 score table for Depression, Anxiety, Stress

3.3 Segregate Label Phase

In this phase, the dataset attributes are categorized into three distinct sets representing physiological disorders: anxiety, depression, and stress, based on predefined criteria. Each disorder set is characterized by five severity levels: Normal (N), Mild (M), Moderate (MD), Severe (S), and Extremely Severe (ES). These severity levels are predefined for the research undertaking, with specific domain ranges established for each disorder set.

To quantify severity of each disorder set, a Weight Score (WS) is computed by summing up data rate points for individual disorder sets across all column cell values, as described by the following equation.

$$W_S = \sum RP^i$$

In this study, the Rate Points (RP) denote the severity levels of each mental disorder set. For instance, instances labeled as "ES" for anxiety, stress, and depression risks are assigned WS values of 20+, 33+, and 28+, respectively. Similarly, severity levels labeled as "S" for these three classes fall within WS ranges of 15–19, 26–33, and 21–27, respectively. Likewise, instances labeled as "MD" have WS ranges of 10–14, 19–25, and 14–20, respectively. The WS values for the "M" class for all risks are within ranges of 8–9, 15–18, and 10–13, respectively. Instances with WS values falling below the specified range are categorized as the "N" class.

3.4 Q-Prioritization Phase

After the data samples undergo preprocessing and labeling, the subsequent Q-Prioritization phase is initiated, responsible for filtering out less significant queries or attributes from the dataset. It is imperative to note that while the preprocessed question set serves as the input, the output of the Q-Prioritization phase is a reduced and optimized question set. Often, not all attributes in the samples contribute equally to the processing and prediction process. The presence of less relevant attributes can impact the overall performance of the predictive model, affecting factors such as latency delay or prediction accuracy. Hence, the detection and elimination of these less relevant features are crucial tasks.

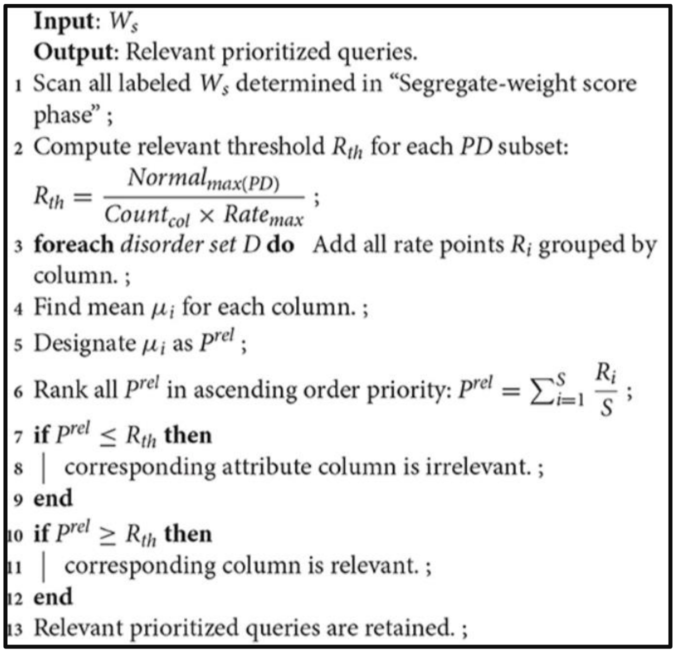
In this pivotal phase, a novel attribute relevance method is employed diligently before the model undergoes training and testing using classifiers. The Weight Score (WS) computed in the preceding phase is meticulously analyzed, and a relevant threshold (Rth) for every physiological disorder set is determined by the following equation:

$$R_{th} = \frac{Normal_{max(PD)}}{Count_{col} \times Rate_{max}}$$

This equation determines the R^{th} for every attribute column of the respective disorder set. For instance, in this study, the computed R^{th} values for anxiety, stress, and depression are 0.5, 1.0, and 0.2, respectively. By computing the cumulative summation of all rate points grouped by individual column names and determining a simple mean average for every column, the priority of attribute columns is established based on the ranking of their mean value in descending order. This priority relevance (P^{rel}) calculation, as shown in equation:

$$P^{rel} = \sum_{i=1}^{S} \frac{R_i}{S}$$

In this study P^{rel} is the priority relevance value to be computed. R_i denotes the rate point for every data sample for a specific disorder set, and “S” is the total samples of data collected. Based on the computation of P^{rel} obtained, those attributes are relevant and are retained. The overall pseudo code of this procedure is depicted in the algorithm:

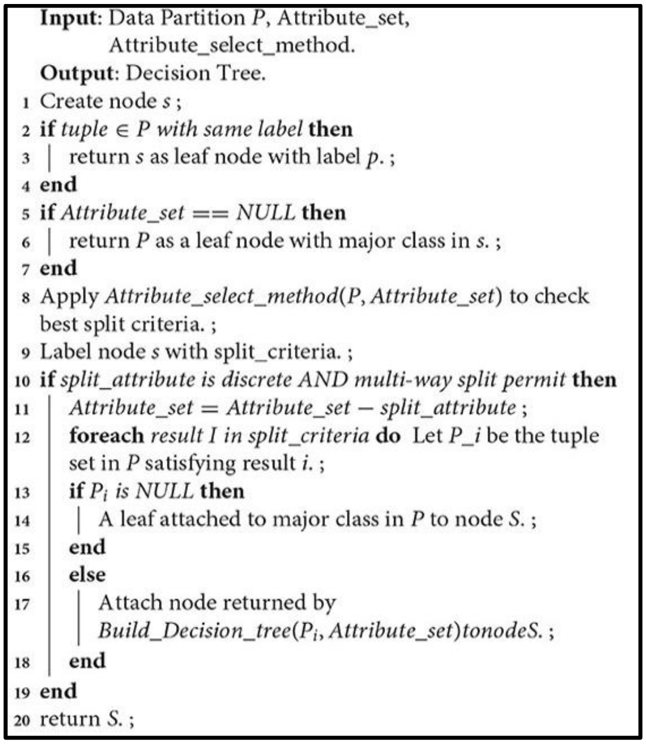


```
Input: $W_s$
Output: Relevant prioritized queries.
Scan all labeled $W_s$ determined in “Segregate-weight score phase” ;
Compute relevant threshold $R_{th}$ for each $PD$ subset:
$R_{th} = \frac{Normal_{max(PD)}}{Count_{col} \times Rate_{max}}$ ;
foreach disorder set $D$ do   Add all rate points $R_i$ grouped by column. ;
Find mean $\mu_i$ for each column. ;
Designate $\mu_i$ as $P^{rel}$ ;
Rank all $P^{rel}$ in ascending order priority: $P^{rel} = \sum_{i=1}^{S} \frac{R_i}{S}$ ;
if $P^{rel} \leq R_{th}$ then
    corresponding attribute column is irrelevant. ;
end
if $P^{rel} \geq R_{th}$ then
    corresponding column is relevant. ;
end
Relevant prioritized queries are retained. ;
```

Algorithm 1: Pseudocode for Q-prioritization phase

3.5 Train-Test Phase

Once the irrelevant attributes in the dataset are eliminated, the dataset is prepared for learning through examples using an appropriate machine intelligence algorithm. Following adequate training, the model can be utilized to predict outcomes based on new data samples. In our research, an improved adaptation of the decision tree algorithm is deployed as the machine learning method. The decision tree serves as a widely utilized supervised approach to solve classification problems, presenting itself as a hierarchical structure with internal nodes corresponding to attributes, branches indicating rules, and leaf nodes representing decision outputs. The decision node and leaf node are the two essential components of a decision tree. Decision nodes are responsible for making decisions, while the outcomes of decisions are determined by leaf nodes. Decisions are made based on the attributes of the dataset. A decision tree predicts the label of a data sample from the root node, comparing its values to the data variable and moving through subsequent nodes based on comparisons. This process continues until it reaches a leaf node. Algorithm as follows:



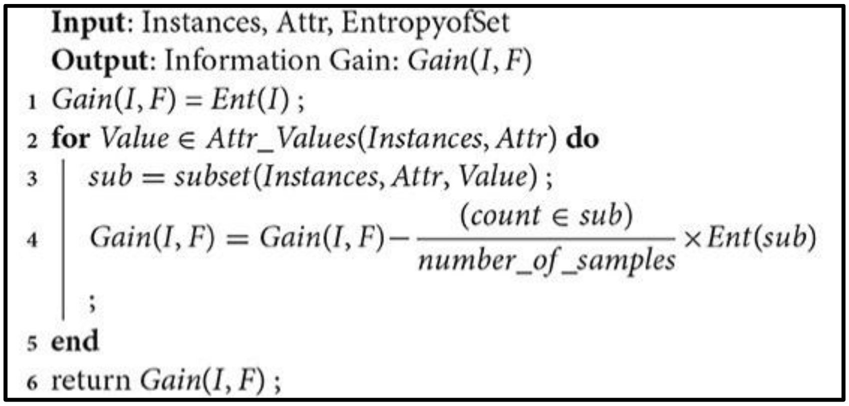
```
Input: Data Partition $P$, Attribute_set,
       Attribute_select_method.
Output: Decision Tree.
Create node $s$ ;
if $tuple \in P$ with same label then
    return $s$ as leaf node with label $p$. ;
end
if $Attribute\_set == NULL$ then
    return $P$ as a leaf node with major class in $s$. ;
end
Apply $Attribute\_select\_method(P, Attribute\_set)$ to check
best split criteria. ;
Label node $s$ with split_criteria. ;
if split_attribute is discrete AND multi-way split permit then
    $Attribute\_set = Attribute\_set - split\_attribute$ ;
    foreach result I in split_criteria do  Let P_i be the tuple
    set in $P$ satisfying result $i$. ;
    if $P_i$ is NULL then
        A leaf attached to major class in $P$ to node $S$. ;
    end
    else
        Attach node returned by
        $Build\_Decision\_tree(P_i, Attribute\_set) tonodeS$. ;
    end
end
return $S$. ;
```

Algorithm 2: Decision tree building from training instances of data partition *P*

In the implementation of a decision tree, a key consideration is selecting the optimal attribute for splitting. This task is achieved through an attribute selection measure, with information gain being a widely used method for this purpose. Information gain assesses the entropy changes resulting from attribute-driven data segmentation, indicating the information an attribute provides about a class. Based on this measure, nodes are split, and the decision tree is constructed. The information Gain (I, F) of an attribute A concerning a set of samples S is denoted by following equation:

$$Gain(I, F) = Ent(I) - \sum_{s \in values(F)} \frac{|I_s|}{|I|} \times Ent(I_s)$$

Where values (F) represent the set of all possible values for attribute F, and Is denotes the subset of I where attribute F has the value s. The pseudocode for information gain is depicted in the algorithm as follows:



```
**Input**: Instances, Attr, EntropyofSet
**Output**: Information Gain: $Gain(I, F)$
$Gain(I, F) = Ent(I)$ ;
**for** $Value \in Attr\_Values(Instances, Attr)$ **do**
    $sub = subset(Instances, Attr, Value)$ ;
    $Gain(I, F) = Gain(I, F) - \frac{(count \in sub)}{number\_of\_samples} \times Ent(sub)$
    ;
**end**
return $Gain(I, F)$ ;
```

Algorithm 3: Pseudocode for Information gain

Entropy in information theory quantifies the purity of a random dataset. For a target feature with m distinct values, its entropy I for m-wise categorization is expressed by:

$$Ent(I) = \sum_{n=1}^{m} -P_i \times log_2 P_n$$

3.5.1 Proposed Balanced Decision Tree Method

Most predictive learning techniques yield optimistic classification performance, often misleading due to uneven data distribution among classes. To address this issue, data sampling methods can be employed to map unbalanced datasets onto more evenly distributed classes. The decision tree, biased toward the majority class, results in higher misclassification rates for the minor class. Implementing oversampling, where existing samples from minor classes are replicated and added to the training set, helps tackle this uneven distribution. By incorporating the oversampling technique, a balanced decision tree model ensures an even distribution of data samples across major and minor classes, significantly enhancing performance.

The improved decision tree model in this study utilizes oversampling to generate a more efficient model. After preprocessing and removing less relevant attributes, the dataset is partitioned into anxiety, stress, and depression sets. Data instances are mapped onto five disorder levels using simple if-then rules, followed by oversampling of the minority class in the training samples. The decision tree algorithm is applied to the resultant oversampled dataset, with the sampling rate incrementing until the maximum accuracy is achieved. This balanced decision tree model effectively detects the presence and predicts the level of mental disorder, ensuring more accurate diagnoses.

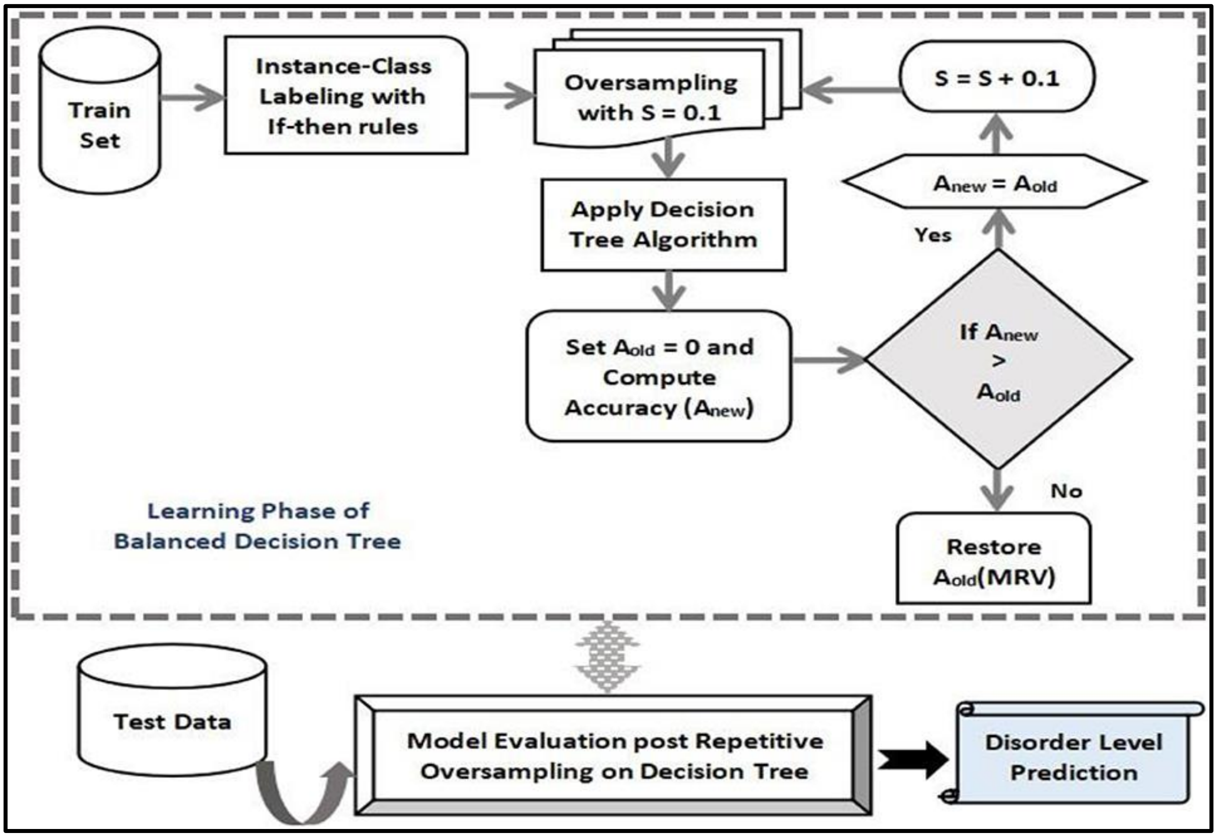

Train Set
Instance-Class Labeling with If-then rules
Oversampling with S = 0.1
S = S + 0.1
Apply Decision Tree Algorithm
A_new = A_old
Yes
Set A_old = 0 and Compute Accuracy (A_new)
If A_new > A_old
No
Learning Phase of Balanced Decision Tree
Restore A_old(MRV)
Test Data
Model Evaluation post Repetitive Oversampling on Decision Tree
Disorder Level Prediction

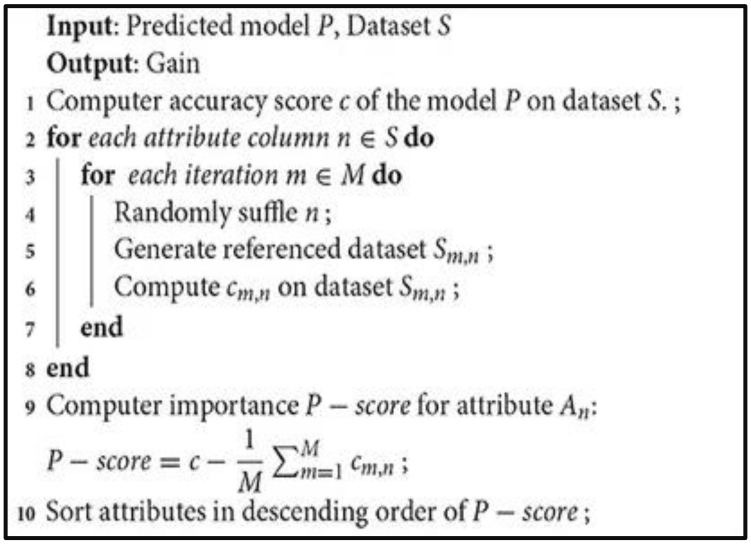
Figure 1: Proposed Balanced decision tree approach

3.6 Prediction Interpret Phase

An essential consideration is determining which attributes significantly influence prediction outcomes, a concern often overlooked by many predictive models, particularly in critical healthcare scenarios. In our research on detecting psychological health risk severity, interpreting predictive results accurately is crucial from both medical experts' and patients' perspectives. To address this, our proposed model integrates a predict-interpret phase, leveraging explainable intelligence facilitated by a reasoning engine. This engine assigns scoring values to attributes based on their individual relevance, aiding in dataset insights and providing a ranking of attribute relevance for mental health risk predictions. Our model employs a novel reasoning engine, incorporating permuted feature importance, contrastive explanation, and counterfactuals methods to enhance explainable intelligence functionality. Permutation feature importance evaluates attribute-target label relationships, while the contrastive explanation method (CEM) offers local interpretations and identifies less relevant features. Additionally, counterfactual interpretation explores feature variations that alter predictions to predetermined outcomes.

3.6.1 Permuted Feature Importance Method

Permuted feature importance calculates attribute significance scores independently of the model, offering a comprehensive understanding of predictive learners. Attributes are ranked based on their influence on predictive decisions. This method assesses attribute prediction ability by measuring the increase in prediction error when attributes are absent. Initially, base error is computed on a trained model. For each attribute, trained data columns are randomly shuffled, and predicted error is determined. The difference between base and shuffled scores determines feature importance. Iterations reduce random shuffling impact, and attributes are ranked based on mean relevance to the model's score. High score relevance attributes are labeled as more significant. The pseudocode is shown in the algorithm, implementation via permutation_importance() function requires a fit model, dataset, and scoring function. Relative importance can be visualized in a bar chart, with longer bars indicating higher relevance.

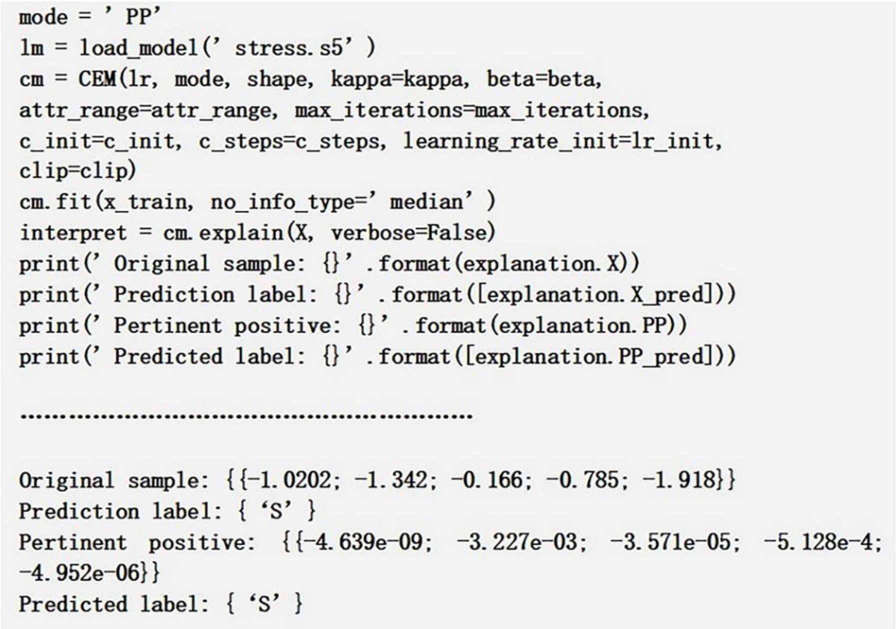


```
Input: Predicted model $P$, Dataset $S$
Output: Gain
Computer accuracy score $c$ of the model $P$ on dataset $S$. ;
for each attribute column $n \in S$ do
    for each iteration $m \in M$ do
        Randomly suffle $n$ ;
        Generate referenced dataset $S_{m,n}$ ;
        Compute $c_{m,n}$ on dataset $S_{m,n}$ ;
    end
end
Computer importance $P - score$ for attribute $A_n$:
$P - score = c - \frac{1}{M} \sum_{m=1}^{M} c_{m,n}$ ;
Sort attributes in descending order of $P - score$ ;
```

Algorithm 4: Pseudocode for permuted feature importance method.

3.6.2 Contrastive Explanation Method

Machine learning models, especially in healthcare, often function as "black boxes," providing predictions without clear explanations. This lack of transparency makes it difficult for medical experts to trust and interpret the model's decisions. Contrastive explanations address this by clarifying why a specific event or prediction occurred in contrast to another possible outcome. This approach is more aligned with human reasoning, as people naturally seek to understand causes by comparing alternatives. A contrastive explanation answers the question, "Why P, rather than Q?" where P is the event being explained (e.g., a model's prediction), and Q is a contrasting alternative. Instead of listing all potential causes, a contrastive explanation focuses on the factors that differentiate P from Q.



```
mode = ' PP'
lm = load_model(' stress.s5' )
cm = CEM(lr, mode, shape, kappa=kappa, beta=beta,
attr_range=attr_range, max_iterations=max_iterations,
c_init=c_init, c_steps=c_steps, learning_rate_init=lr_init,
clip=clip)
cm.fit(x_train, no_info_type=' median' )
interpret = cm.explain(X, verbose=False)
print(' Original sample: {}' .format(explanation.X))
print(' Prediction label: {}' .format([explanation.X_pred]))
print(' Pertinent positive: {}' .format(explanation.PP))
print(' Predicted label: {}' .format([explanation.PP_pred]))

....................................................

Original sample: {{-1.0202; -1.342; -0.166; -0.785; -1.918}}
Prediction label: { 'S' }
Pertinent positive: {{-4.639e-09; -3.227e-03; -3.571e-05; -5.128e-4;
-4.952e-06}}
Predicted label: { 'S' }
```

Figure 2: Pertinent positive sample example.

3.7 Our Implementation & Findings

3.7.1 Dataset Overview

In our project, we utilized the standard DASS 42 questionnaire dataset for performance evaluation purposes. This dataset serves as a fundamental tool for assessing various aspects of mental health, including depression, anxiety, and stress levels. Notably, our dataset represents a significant expansion in scale compared to previous studies, encompassing a total of 33,281 unique responses. This substantial increase in sample size provides a more comprehensive and robust foundation for our analysis and findings. By leveraging this extensive dataset, we aim to achieve more accurate insights into the factors influencing mental health and develop more effective strategies for addressing related challenges.

3.7.2 Data Preprocessing

As part of our project execution, we rigorously implemented the Transform-Encode Phase and Segregate-Label Phase, adhering closely to the methodology outlined in earlier discussions. During the Transform-Encode Phase, we processed the raw data from the DASS 42 questionnaire dataset, transforming it into a format suitable for analysis. This involved encoding categorical variables, handling missing data, and standardizing features as necessary to ensure consistency and accuracy in our subsequent analyses. Subsequently, in the Segregate-Label Phase, we meticulously organized the dataset into appropriate subsets based on specific criteria, enabling us to effectively label and categorize the data for targeted analysis. By meticulously following these methodological steps, we established a solid foundation for our research, ensuring that our subsequent analyses are based on high-quality, well-prepared data.

3.7.3 Feature Selection

In our project, we delved into various techniques for feature selection to optimize the performance of our models. We explored methods such as Q-prioritization, feature variance analysis, and Information gain analysis to identify the most relevant features from the DASS 42 questionnaire dataset. Despite our initial efforts and thorough exploration of these techniques, we encountered a challenge where feature selection resulted in an overall reduction in model performance. This unexpected outcome prompted us to reassess our approach and delve deeper into understanding the underlying reasons behind this phenomenon. By acknowledging and documenting this setback, we aim to provide transparency in our methodology and pave the way for further investigation and refinement in future iterations of our project.

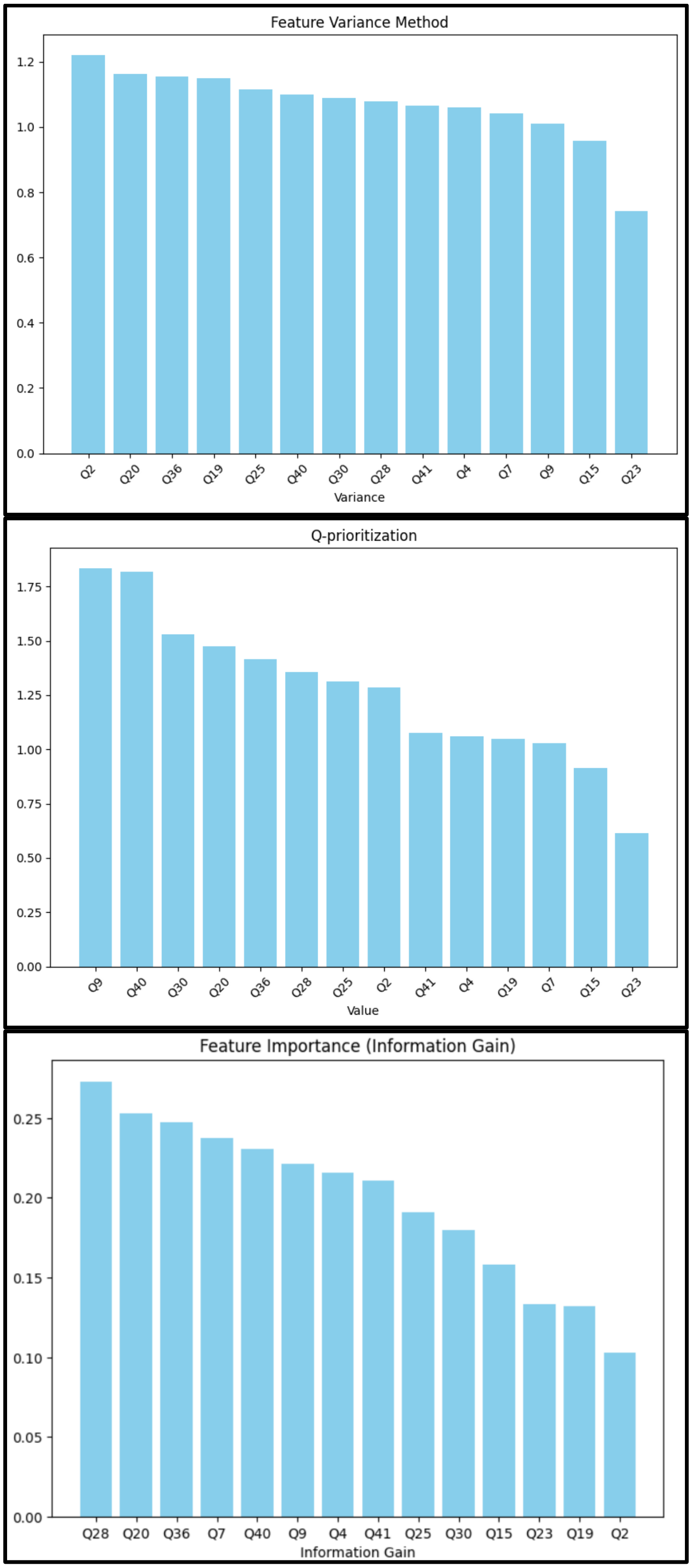
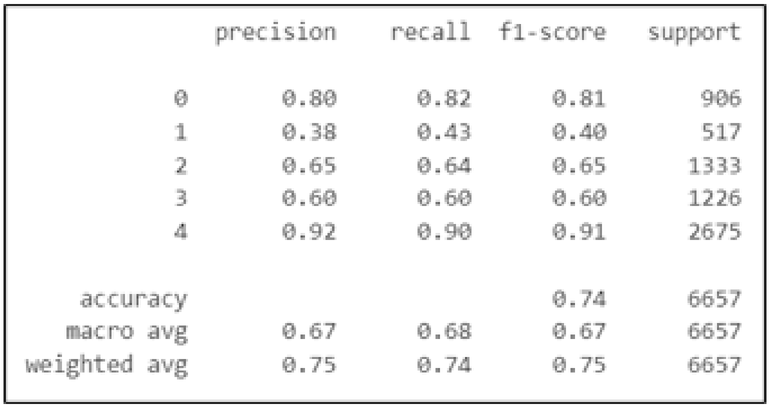


Figure 3: Feature Selection for Anxiety Data

3.7.4 Model Selection and Training-Testing

We prioritized the selection of appropriate models and the optimization of their performance through meticulous training and testing procedures. To achieve this, we utilized GridSearchCV, a powerful tool for hyperparameter tuning, specifically in the decision tree model. GridSearchCV allowed us to systematically explore a range of hyperparameter combinations, such as tree depth and minimum sample split, to identify the configuration that maximizes model performance. By employing this method, we aimed to fine-tune our decision tree model and enhance its ability to accurately classify and predict mental health outcomes based on the DASS 42 questionnaire dataset. Through this rigorous approach to model selection and hyperparameter optimization, we sought to ensure that our final model delivers reliable and robust results, ultimately contributing to the effectiveness of our project's objectives.

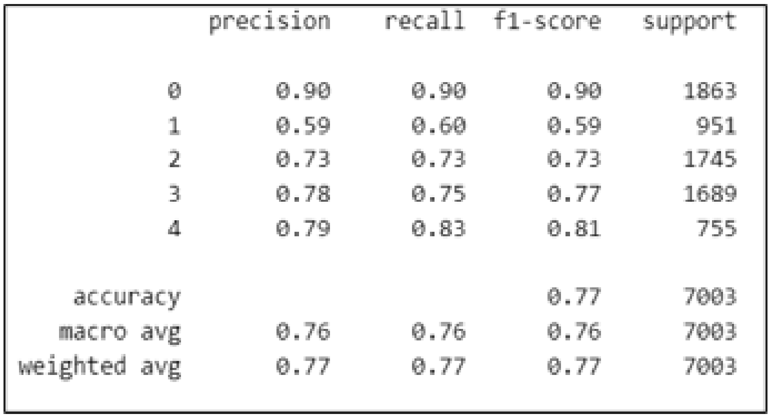


```
              precision    recall  f1-score   support

           0       0.80      0.82      0.81       906
           1       0.38      0.43      0.40       517
           2       0.65      0.64      0.65      1333
           3       0.60      0.60      0.60      1226
           4       0.92      0.90      0.91      2675

    accuracy                           0.74      6657
   macro avg       0.67      0.68      0.67      6657
weighted avg       0.75      0.74      0.75      6657
```

Anxiety

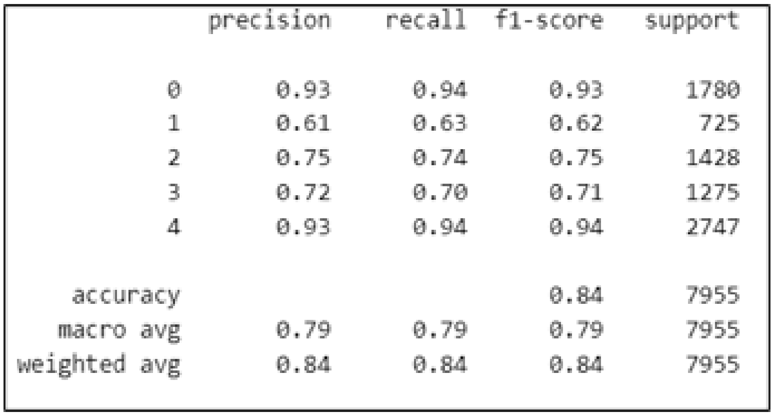


```
              precision    recall  f1-score   support

           0       0.90      0.90      0.90      1863
           1       0.59      0.60      0.59       951
           2       0.73      0.73      0.73      1745
           3       0.78      0.75      0.77      1689
           4       0.79      0.83      0.81       755

    accuracy                           0.77      7003
   macro avg       0.76      0.76      0.76      7003
weighted avg       0.77      0.77      0.77      7003
```

Stress



```
              precision    recall  f1-score   support

           0       0.93      0.94      0.93      1780
           1       0.61      0.63      0.62       725
           2       0.75      0.74      0.75      1428
           3       0.72      0.70      0.71      1275
           4       0.93      0.94      0.94      2747

    accuracy                           0.84      7955
   macro avg       0.79      0.79      0.79      7955
weighted avg       0.84      0.84      0.84      7955
```

Depression

Figure 4: Classification Reports for Decision Tree Model

3.7.5 Interpretation Phase

During the interpretation phase of our project, we employed advanced techniques to gain insights into the decision-making process of our models. Specifically, we utilized two prominent methods: CEM (Counterfactual Explanations Method) and the Permuted Feature Importance Method.

CEM provided us with valuable counterfactual explanations, allowing us to understand how changes in input features would alter the model's predictions. By generating these counterfactual scenarios, we gained deeper insights into the underlying mechanisms driving the decision tree model's outputs, thereby enhancing our understanding of its behavior and predictive patterns.

Additionally, we leveraged the Permuted Feature Importance Method to assess the significance of each feature in influencing the model's predictions. This method involved systematically permuting the values of individual features and observing the resulting changes in model performance. By quantifying the impact of each feature on the model's accuracy, we could prioritize features based on their importance in the decision-making process.

Through the combined use of CEM and the Permuted Feature Importance Method, we aimed to elucidate the inner workings of our decision tree model, uncovering key insights that could inform future research directions and real-world applications in mental health assessment and intervention strategies.

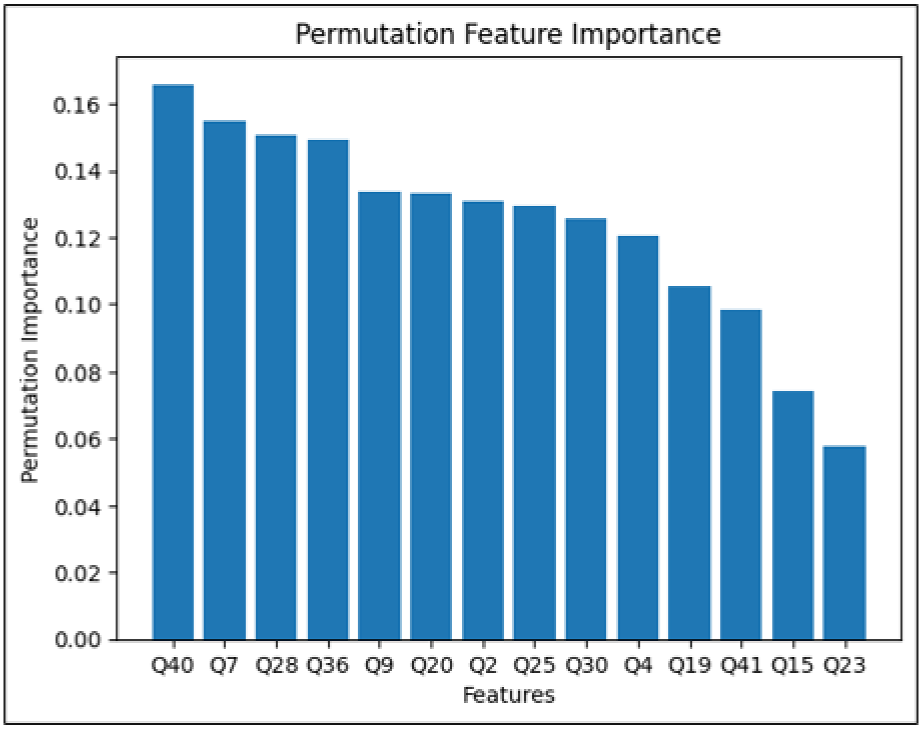


Figure 5: Decision Tree Model (Anxiety Data)

3.8 Limitations

Despite our efforts, we encountered several limitations in our project:

Model Accuracy Discrepancies: The accuracy of our models for predicting anxiety, stress, and depression levels did not meet our initial expectations. Further analysis is necessary to investigate the factors contributing to this discrepancy and refine our models for better performance.

Q-prioritization Impact: Our exploration of Q-prioritization for feature selection resulted in a reduction in model accuracy. It is essential to understand the reasons behind this outcome, as it may inform adjustments to our feature selection strategy or the identification of alternative methods.

Data Complexity: The data used in our project was of zero complexity, with well-defined boundaries and no overlapping categories. As a result, traditional ML models like SVM, which use lines to divide regions in the data, could achieve 100% accuracy.

Due to these limitations, we chose to focus on another area of work: mental disorder detection, primarily stress detection from social media posts using NLP. This approach leverages the complexity and nuances in textual data to develop more sophisticated models for mental health assessment.

4. Part Two Work

4.1 Dreaddit Dataset Overview

The dataset used for this research comprises Reddit posts, focusing on identifying stress indicators from social media content. It is built upon Reddit data, as the platform is characterized by lengthy, detailed posts that lend themselves well to studying complex phenomena like stress. The dataset content is basically text from Reddit posts having size of 187,444 posts, spanning from January 1, 2017, to November 19, 2018, where the average post length is 420 tokens (significantly longer than microblog data like Twitter). Here’s a detailed overview of the dataset, including its key components and features.

Key Dataset Details

- **Corpus**: The dataset includes the body of Reddit posts, the name of the subreddit, and various other features. This provides a rich source of text data for analysis.
- **Labels**: Each post is labeled as either "Stress" (1) or "Not Stress" (0). These labels are crucial for training and evaluating stress detection models.
- **Confidence**: An agreement threshold for data labeling, ensuring that the labels assigned to the posts are reliable and consistent. This confidence metric helps in maintaining the quality of the dataset.
- **Total Posts**: The dataset contains a total of 3,553 posts.
- **Posts Statistics**: Stressful Segments: 1,857 (52.3%), Non-Stressful Segments: 1,696 (47.7%). This indicates a significant proportion of the data is focused on stress detection.

Features Extracted from the Dataset

The features used for stress detection from Reddit posts are categorized into three main groups: **Lexical**, **Syntactic**, and **Social Media** features. Each category includes specific types of data derived from the text and context of the posts. Here’s an overview of each feature category:

- **Lexical features**: These include the average, maximum, and minimum scores for pleasantness, activation, and imagery from the Dictionary of Affect in Language (DAL) (Whissel, 2009); the full suite of 93 LIWC features; and sentiment calculated using the Pattern sentiment library (Smedt and Daelemans, 2012).

- **Syntactic features**: These encompass part-of-speech unigrams and bigrams, the Flesch-Kincaid Grade Level, and the Automated Readability Index.

- **Social media features**: These cover the UTC timestamp of the post; the ratio of upvotes to downvotes (upvote ratio); the net score of the post (karma), calculated by Reddit as the number of upvotes minus the number of downvotes; and the total number of comments in the entire thread under the post.

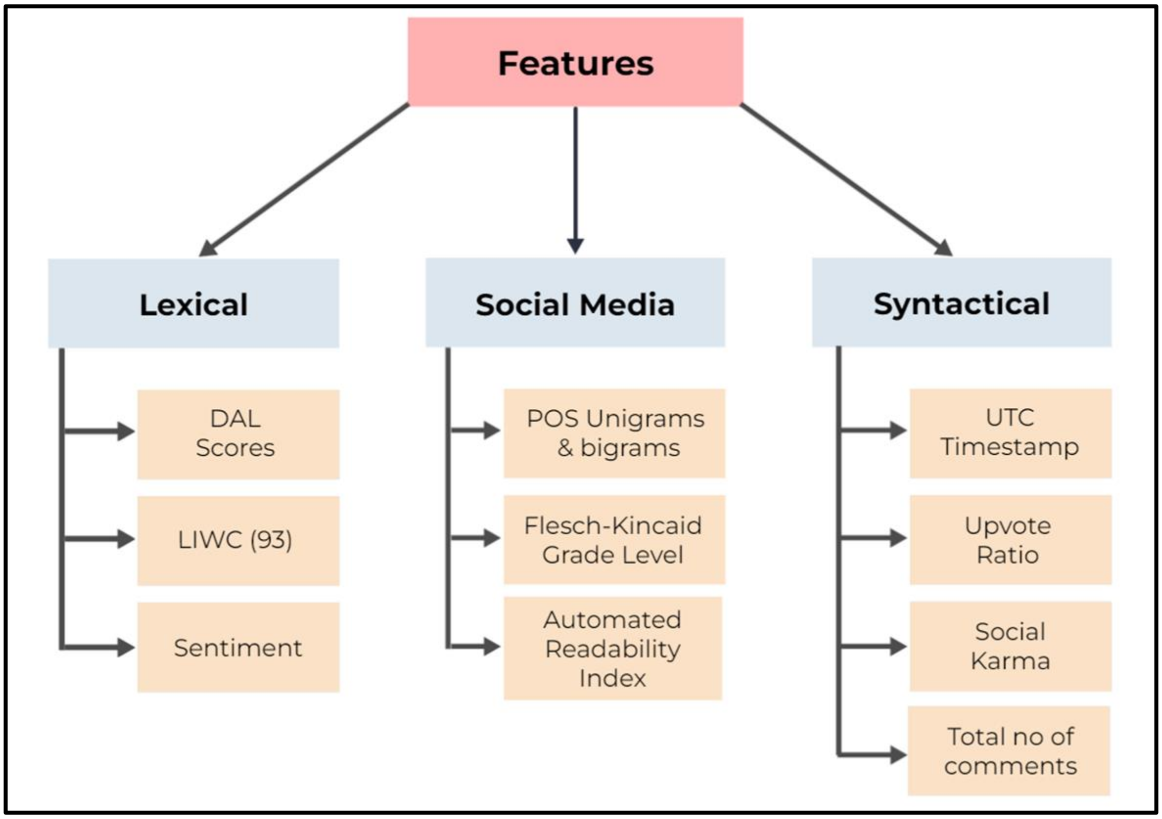


Figure 6: Extracted features from dataset

4.2 NLP Pipeline

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) focused on the interaction between computers and humans through natural language. The primary goal of NLP is to enable computers to understand, interpret, and generate human language in a way that is both meaningful and useful. Applications of NLP include sentiment analysis, language translation, chatbots, and more. The NLP pipeline refers to the sequence of steps involved in processing and analyzing natural language data. This pipeline transforms raw text into a format that can be used for machine learning models. There are multiple steps that are followed in NLP pipeline. In this project the steps include:

i. Dataset Collection
ii. Post Body (Text)
iii. Text Preprocessing
iv. Word Embeddings
v. Feature Engineering
vi. Selected Features + Embedding
vii. Train Models
viii. Evaluation

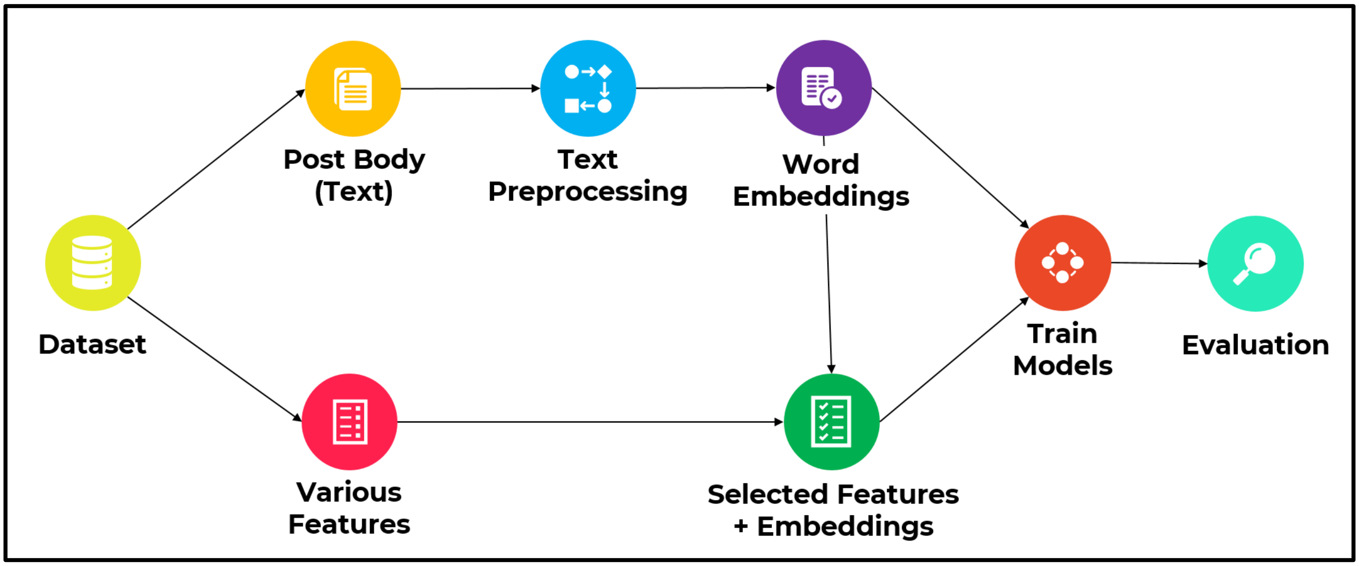


Figure 7: Implemented NLP Pipeline

4.3 Dataset Collection

The initial step involves gathering data that will be used for training and evaluating the NLP models. In the provided research, the dataset is collected from Reddit, a social media platform where users post in topic-specific communities called subreddits. The data includes posts from subreddits likely to contain discussions about stress, such as those focused on interpersonal conflict and mental illness. This dataset is then annotated for stress using Amazon Mechanical Turk, resulting in a labeled corpus of 190K posts and 3.5K segments. The dataset is available **here** (attach link).

4.4 Post Body (Text)

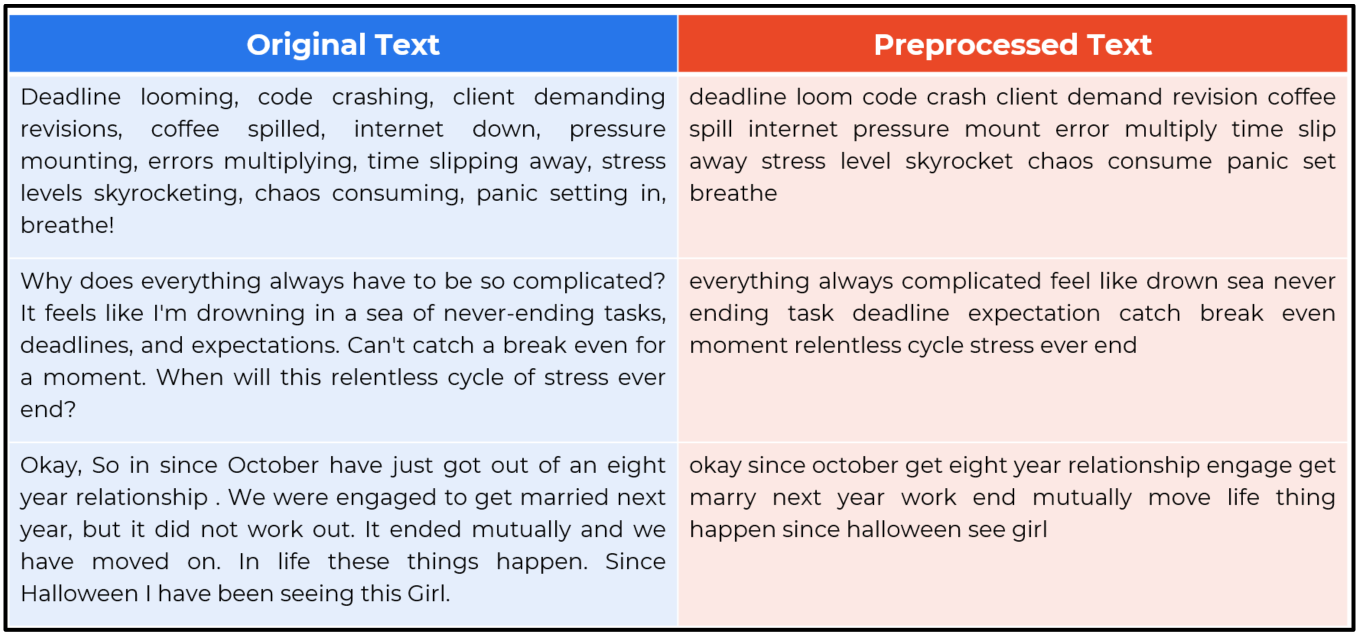
In this step from the dataset the comments are collected which is basically text content which will be further gone to preprocessing step.

4.5 Text Preprocessing

To clean and prepare the text data for analysis by removing noise and standardizing the format. Text preprocessing involves several steps:

- **Fix Contractions**: Expand contractions to their full forms, such as "don't" to "do not."
- **Lowercasing**: Convert all characters in the text to lowercase to ensure uniformity.
- **Replace Emojis**: Substitute emojis with their corresponding text descriptions or meanings.
- **Remove HTML Tags**: Strip out any HTML tags from the text to clean the content.
- **Remove URLs**: Delete all URLs to remove unnecessary links from the text.
- **Remove Numbers**: Eliminate all numerical digits to focus on textual content.
- **Remove Punctuations**: Remove punctuation marks to simplify the text.
- **Chat Word Replacement**: Replace slang, abbreviations, and chat-specific words with their formal equivalents.
- **Tokenization**: Split the text into individual words or tokens to facilitate analysis.
- **Remove Stop words**: Discard common, non-informative words like "and" "the," "is," etc.
- **Stemming**: Reduce words to their base or root form by removing suffixes (e.g., "agreed" to "agre").
- **Lemmatization**: Convert words to their base dictionary form, considering the context (e.g., "agreed" to "agree"). In our work, we used lemmatization instead of stemming to maintain contextual integrity, as we employed many pre-trained embeddings that benefit from preserving the original word forms.

These preprocessing steps ensure that the text data is clean and in a consistent format for further processing. The following table shows the result of preprocessing on different inputs.



Original Text	Preprocessed Text
Deadline looming, code crashing, client demanding revisions, coffee spilled, internet down, pressure mounting, errors multiplying, time slipping away, stress levels skyrocketing, chaos consuming, panic setting in, breathe!	deadline loom code crash client demand revision coffee spill internet pressure mount error multiply time slip away stress level skyrocket chaos consume panic set breathe
Why does everything always have to be so complicated? It feels like I'm drowning in a sea of never-ending tasks, deadlines, and expectations. Can't catch a break even for a moment. When will this relentless cycle of stress ever end?	everything always complicated feel like drown sea never ending task deadline expectation catch break even moment relentless cycle stress ever end
Okay, So in since October have just got out of an eight year relationship . We were engaged to get married next year, but it did not work out. It ended mutually and we have moved on. In life these things happen. Since Halloween I have been seeing this Girl.	okay since october get eight year relationship engage get marry next year work end mutually move life thing happen since halloween see girl

Table 3: Text Preprocessing Example

4.6 Word Embedding

Word embedding is a technique in natural language processing (NLP) where words or phrases from a vocabulary are mapped to vectors of real numbers. This process allows words with similar meanings to have similar representations in the vector space. The primary purpose of word embedding is to capture the semantic relationships between words, which facilitates various NLP tasks such as text classification, sentiment analysis, and machine translation. In the research work the embeddings that are used were: pretrained Word2Vec and domain Word2Vec. We have additionally performed multiple embeddings. The utilized word embedding techniques are:

i. Bag of Words (BoW)
ii. TF-IDF
iii. Domain Word2Vec
iv. Pretrained Word2Vec
v. Fasttext
vi. BERT

These embeddings are used to represent the textual data numerically, facilitating the development of models for stress detection. Below, we describe each of these embedding techniques and their application in our work.

Bag Of Words:

The Bag of Words model is a simple and widely used method for text representation. It converts text into a vector of word counts, ignoring grammar and word order but preserving the frequency of words. We have performed operations of multiple models which includes the following: Multinomial Naive Bayes, Gaussian Naive Bayes, Bernoulli Naive Bayes, Support Vector Classifier (SVC), Random Forest Classifier, Logistic Regression, XGB Classifier, Artificial Neural Network (ANN).

TF-IDF:

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. It reflects how frequently a word appears in a document while offsetting its frequency in the entire corpus. TF-IDF is used to create vectors that consider both the frequency and the uniqueness of words. Using this embedding we have performed the same models as there in BoW.

Domain Word2Vec:

Word2Vec is a neural network-based word embedding technique developed by Google. It generates dense vector representations of words by training on large corpora. Domain Word2Vec embeddings are specifically trained on the collected Reddit data, making them domain-specific and more relevant for the task of stress detection. Apart from the previously mentioned models LSTM, Bi-LSTM, Gated Recurrent Unit Network (GRU) are also utilized.

Pretrained Word2Vec:

Pretrained Word2Vec embeddings are trained on a large, generic corpus, such as Google News. These embeddings capture a broad range of linguistic patterns and relationships. Pretrained Word2Vec embeddings are used to leverage extensive pre-existing linguistic knowledge. Multinomial Naive Bayes, Gaussian Naive Bayes, Bernoulli Naive Bayes, Support Vector Classifier (SVC), Random Forest Classifier, Logistic Regression, XGB Classifier, Artificial Neural Network (ANN) were used with it.

BERT:

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based model developed by Google. It processes text bidirectionally, capturing context from both directions, and is highly effective for various NLP tasks. BERT embeddings are used to leverage deep contextual understanding of words. In the research it is used as a standalone embedding technique.

Fasttext:

Fasttext is an open-source library developed by Facebook's AI Research (FAIR) lab for efficient text representation and classification in natural language processing (NLP) tasks. It is designed to handle large volumes of text data efficiently and is particularly useful for tasks such as text classification, sentiment analysis, and word embedding.

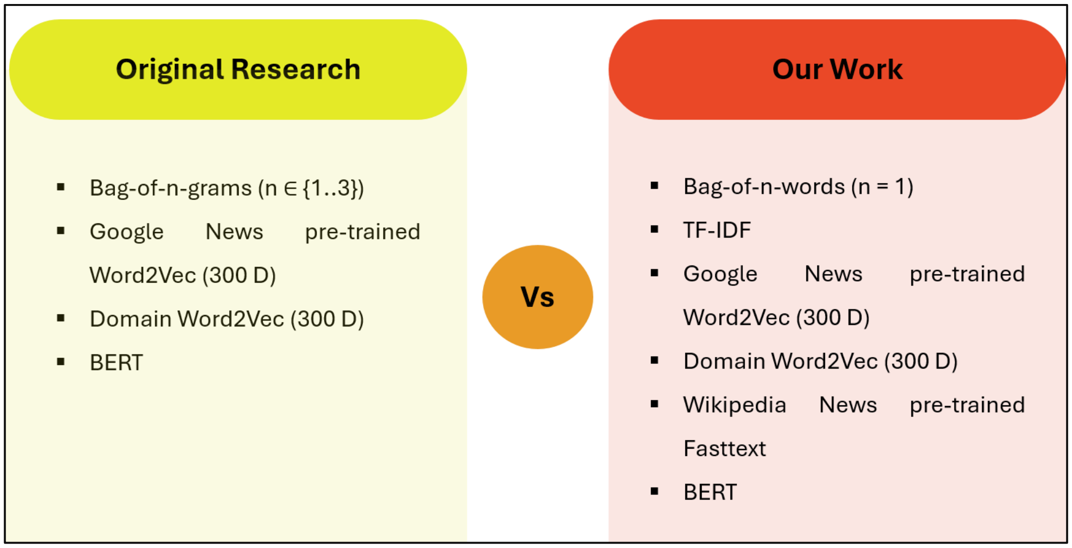
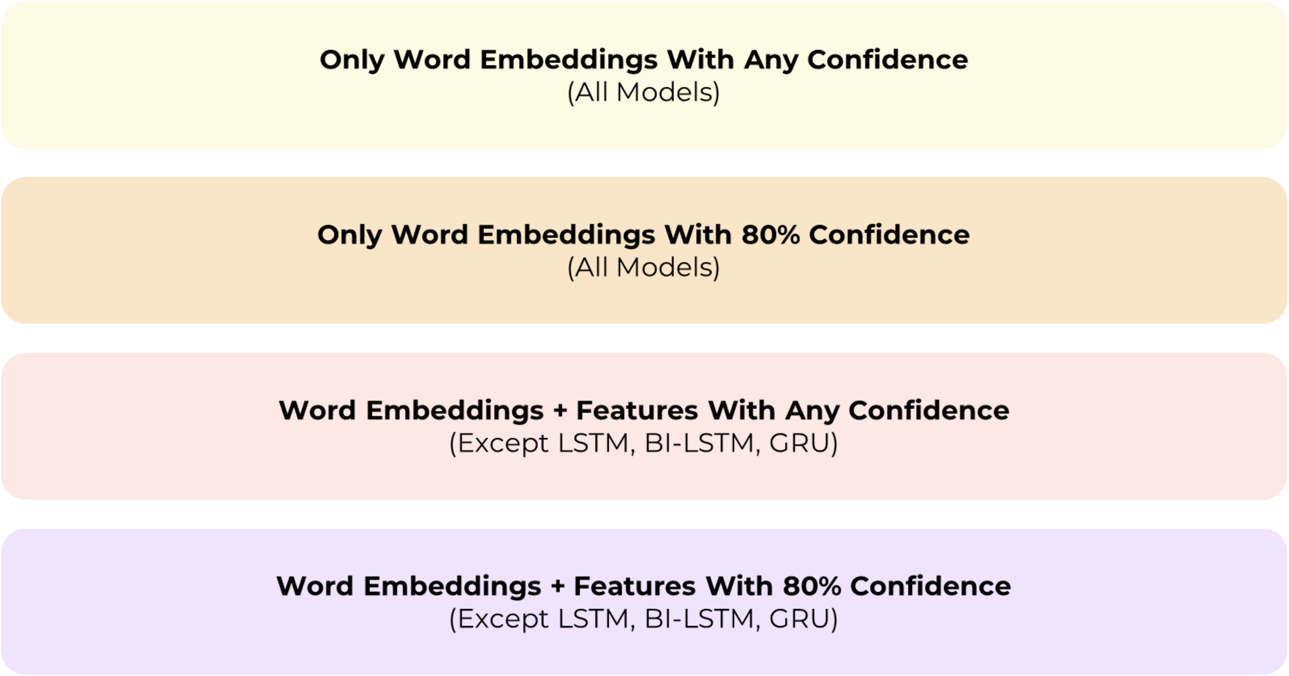

Original Research
▪ Bag-of-n-grams (n ∈ {1..3})
▪ Google News pre-trained Word2Vec (300 D)
▪ Domain Word2Vec (300 D)
▪ BERT
Vs
Our Work
▪ Bag-of-n-words (n = 1)
▪ TF-IDF
▪ Google News pre-trained Word2Vec (300 D)
▪ Domain Word2Vec (300 D)
▪ Wikipedia News pre-trained Fasttext
▪ BERT

Figure 8: Original Research vs Our Work word embeddings

4.7 Model Experimentation

The research paper employs a variety of models, both traditional machine learning and neural network-based, to analyze and detect stress in Reddit posts. The non-neural models used in the research include Support Vector Machines (SVMs), logistic regression, Naive Bayes, Perceptron, and decision trees. For neural models, they utilized a two-layer bidirectional Gated Recurrent Neural Network (GRNN) and a

Convolutional Neural Network (CNN). The input representations experimented with include bag-of-n-grams, Google News pre-trained Word2Vec embeddings, domain-specific Word2Vec embeddings, and BERT embeddings. In comparison, the models used in our work include traditional machine learning models such as SVMs, logistic regression, different types of Naive Bayes (Multinomial, Bernoulli, Gaussian), and random forest, along with gradient boosting models. For neural network approaches, we used Artificial Neural Networks (ANNs), LSTM, Bi-LSTM, and GRU. Multiple models have been used with different embeddings and different confidence. The work is focused on two parts: one in which there is only text and no features and another text + features. And both methods have two different confidence levels: any confidence and 80% confidence.



Only Word Embeddings With Any Confidence
(All Models)

Only Word Embeddings With 80% Confidence
(All Models)

Word Embeddings + Features With Any Confidence
(Except LSTM, BI-LSTM, GRU)

Word Embeddings + Features With 80% Confidence
(Except LSTM, BI-LSTM, GRU)

Figure 9: Model Experimentation Setup

4.7.1 Models Used in Our Work

Traditional Machine Learning

Support Vector Machines (SVMs): SVMs are supervised learning models that analyze data for classification and regression analysis. They work by finding the hyperplane that best separates the classes in the feature space, maximizing the margin between the data points of different classes. We have used different parameters with different embeddings. The main changes lie in the kernel which was used and the Regularization (C). The kernels that are used: linear, sigmoid, polynomial etc. And the C value ranges from 0.1 to 100.

Logistic Regression: Logistic regression is a statistical model used for binary classification. It models the probability of a certain class or event existing, such as whether an email is spam or not spam, using a logistic function to predict binary outcomes. It is being used with max_iter and solver parameter where max_iter sets the maximum number of iterations the algorithm will run to converge, and solver describes the optimization algorithm. Stochastic Average Gradient (SAG) Algorithm is mainly used as the solver.

Naive Bayes (Multinomial, Bernoulli, Gaussian): Naive Bayes classifiers are a family of simple probabilistic classifiers based on Bayes' theorem. The Multinomial Naive Bayes is suited for discrete data, the Bernoulli Naive Bayes is for binary/boolean features, and the Gaussian Naive Bayes is for continuous data assuming a Gaussian distribution. Here the parameters are used with default parameters. In most cases the models ran with default parameters, but they are tuned with alpha, which is the smoothing parameter used to handle zero probabilities.

Random Forest: Random Forest is an ensemble learning method for classification and regression that constructs multiple decision trees during training and outputs the mode of the classes or mean prediction of the individual trees to improve predictive accuracy and control over-fitting and used the default parameters for the work. It is used with default as well as multiple parameters which includes n_estimators that specify the number of trees in the forest, while min_samples_split determines the minimum number of samples required to split an internal node.

Gradient Boosting: Gradient Boosting is another ensemble technique that builds models sequentially, with each new model correcting errors made by the previous ones. It combines models minimize the overall prediction error, where it’s used with default parameters. It is used with default as well as tuning parameters such as ‘learning_rate’ controls the contribution of each tree to the final model, ‘max_depth’ sets the maximum depth of each tree, ‘n_estimators’ specify the number of boosting stages or trees to be added, and ‘subsample’ denotes the fraction of samples used to fit each tree.

Neural Networks

Artificial Neural Networks (ANN): ANNs consist of interconnected groups of artificial neurons that process information using a connectionist approach. In our work, we used 1 or 2 dense layers to capture complex patterns in the data by learning weights during training and used different dropout rates from 0 to 0.5.

LSTM (Long Short-Term Memory): LSTM is a type of recurrent neural network (RNN) capable of learning long-term dependencies. It is particularly effective for tasks involving sequential data, such as time series analysis and natural language processing, by using gates to control the flow of information. We used a single layer neuron used with different dropout rate ranges from 0 to 0.4.

Bi-LSTM (Bidirectional LSTM): Bi-LSTM is an extension of LSTM that processes data in both forward and backward directions. This bidirectional approach allows the model to capture context from both past and future states, enhancing its understanding of the sequence. Used with single layer neurons, spatial dropout rate and dropout rate.

GRU (Gated Recurrent Unit): GRU is another type of RNN that, like LSTM, is designed to handle sequential data and capture long-term dependencies. GRUs have a simpler architecture with fewer parameters compared to LSTMs, making them more efficient while achieving similar performance. The parameters for models were GRU, dropout and spatial dropout as well. GRU is used with 64 or 128 value and dropout and spatial dropout rate with value 0.2.

4.8 Results & Findings

4.8.1 Only Text & Any Confidence

On the diagram we have shown the results for the best outcomes in this scenario. The highest F1-score is achieved by the Fasttext embedding technique using an Artificial Neural Network (ANN) model, scoring 76.86. This indicates that the combination of Fasttext embeddings and ANN is particularly effective for text classification tasks. Pretrained Word2Vec with ANN and BERT also show strong performance, with F1-scores of 73.91 and 74.44, respectively. Traditional methods like TF-IDF with Random Forest and Bag of Words with BNB still perform reasonably well but are outperformed by more advanced embedding techniques and models. Overall, leveraging advanced embeddings like

Fasttext and Pretrained Word2Vec with neural network models provides significant improvements in classification performance.

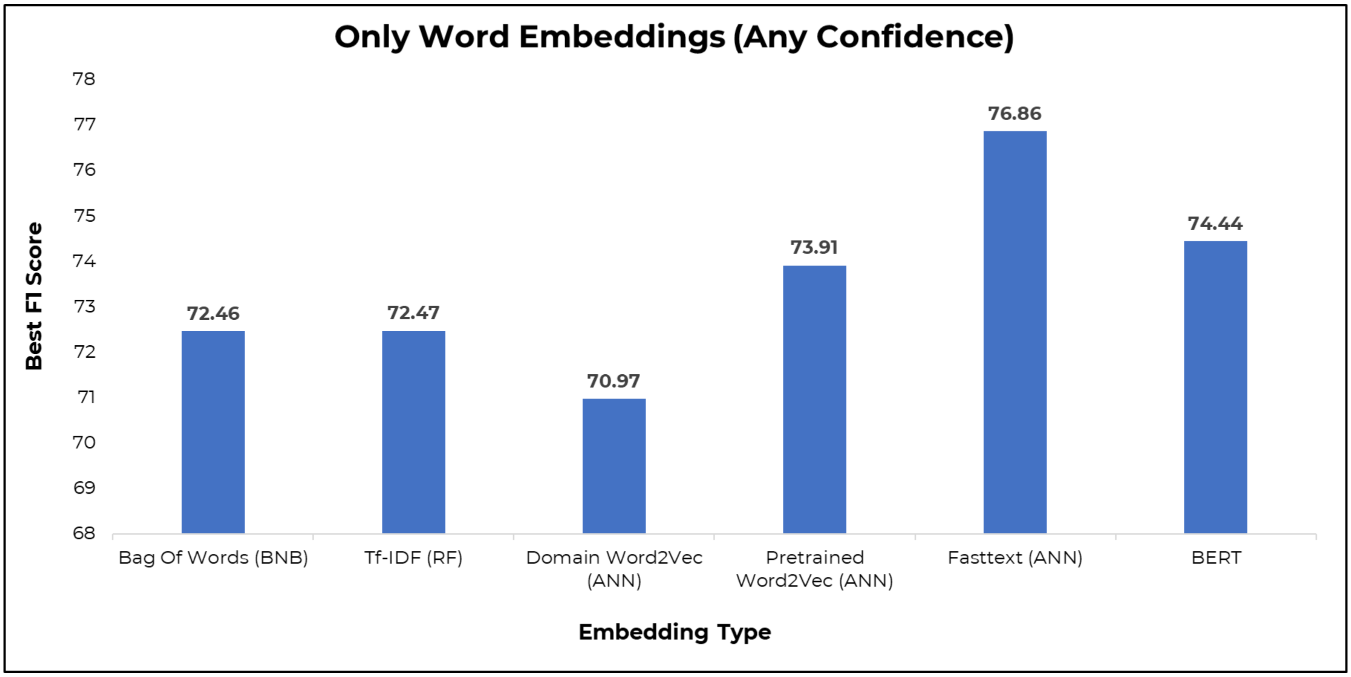


Figure 10: Word Embeddings with Any confidence

4.8.2 Only Text & 80% Confidence

As the results are depicted, the bar graph displays the best F1-scores achieved using different word embedding techniques with 80% confidence intervals. Fasttext embeddings with an Artificial Neural Network (ANN) attained the highest F1-score of 82.62, followed closely by Pretrained Word2Vec embeddings with an ANN at 81.38. TF-IDF with ANN and Domain-specific Word2Vec with a Bidirectional Long Short-Term Memory (Bi-LSTM) also performed well, with scores of 79.99 and 79.5, respectively. Bag of Words with Bernoulli Naive Bayes (BNB) and BERT embeddings showed lower performance, scoring 78.75 and 78.4, respectively. These results indicate that among the tested embeddings, Fasttext and Pretrained Word2Vec with ANN yield the best performance in this context.

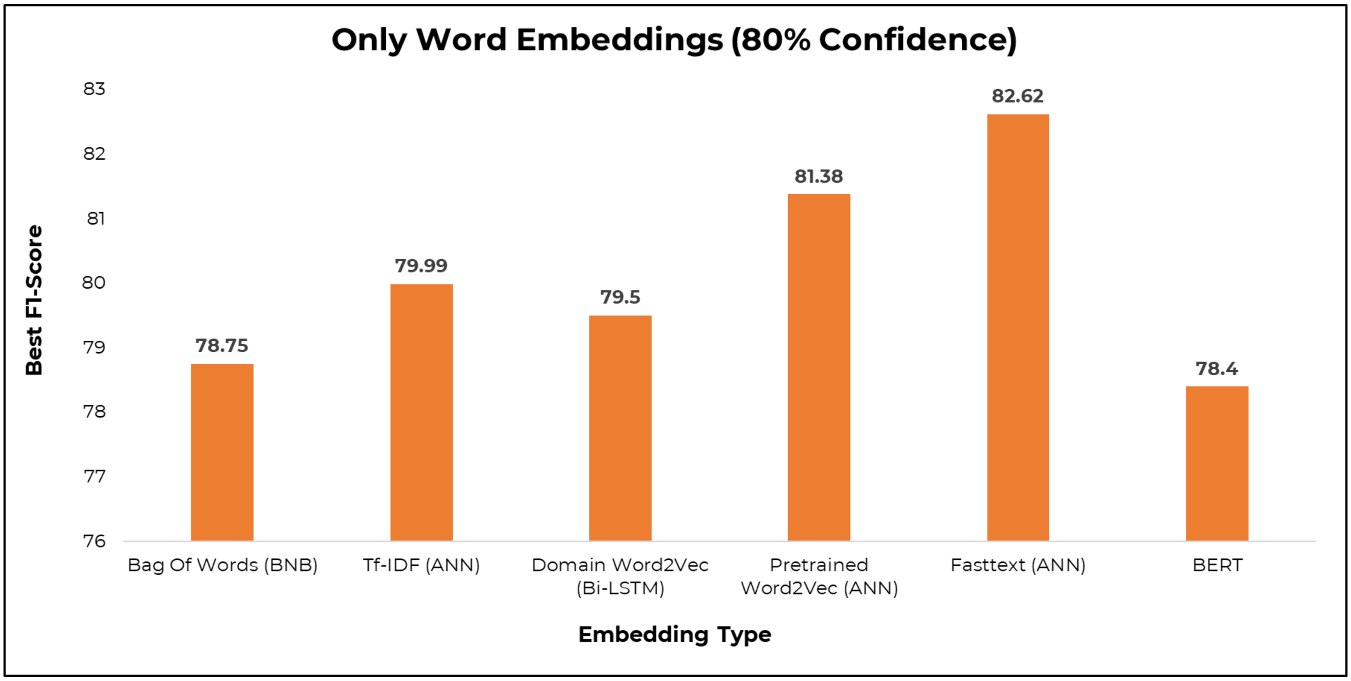


Figure 11: Word embeddings with 80% confidence

4.8.3 Word Embedding + Features & Any Confidence

The graph shows the best F1-scores achieved using various word embedding techniques combined with additional features, evaluated with any confidence level. Pretrained Word2Vec embeddings with an Artificial Neural Network (ANN) attained the highest F1-score of 78.34, followed closely by TF-IDF with ANN and Bag of Words with ANN, both scoring 78.27. Fasttext with ANN also performed well, with a score of 77.81. Domain-specific Word2Vec embeddings with ANN had the lowest performance, scoring 76.94. These results indicate that the combination of pretrained embeddings and additional features generally improves performance, with Pretrained Word2Vec providing the best results in this context.

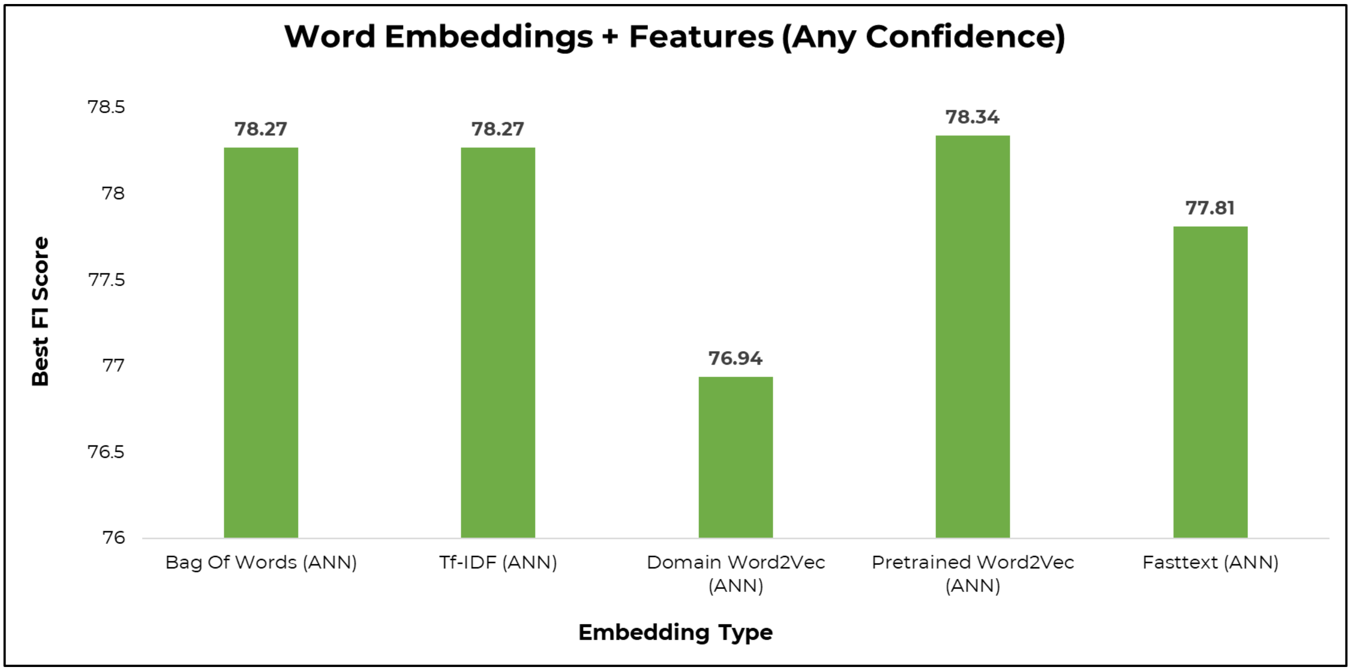


Figure 12: Word embeddings with features and any confidence

4.8.4 Word Embedding + Features & 80% Confidence

In this scenario the graph illustrates the best F1-scores achieved using different word embedding techniques combined with additional features, evaluated with 80% confidence intervals. Bag of Words with an Artificial Neural Network (ANN) achieved the highest F1-score of 83.47, followed by Pretrained Word2Vec embeddings with ANN, which scored 83.03. Domain-specific Word2Vec with ANN, Fasttext with ANN, and TF-IDF with ANN also performed well, with scores of 82.87, 82.42, and 82.56, respectively. These results indicate that incorporating additional features and using Bag of Words or Pretrained Word2Vec embeddings with ANN provides the highest performance in this context.

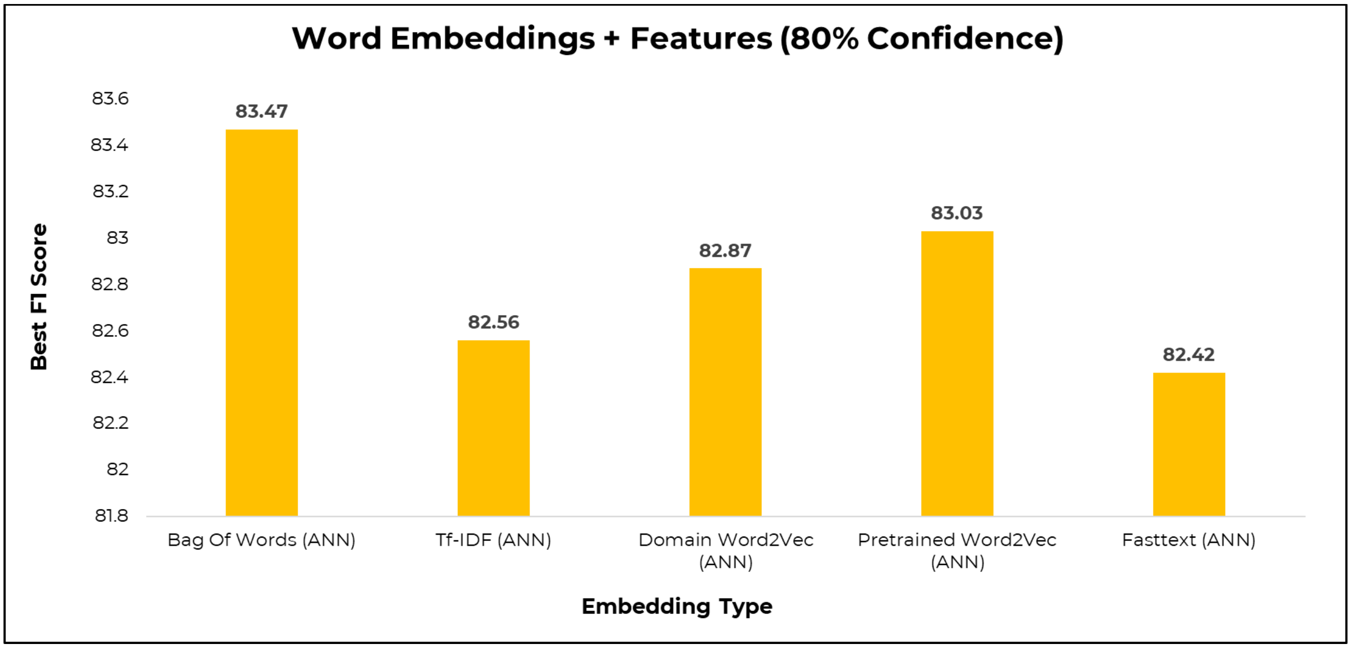
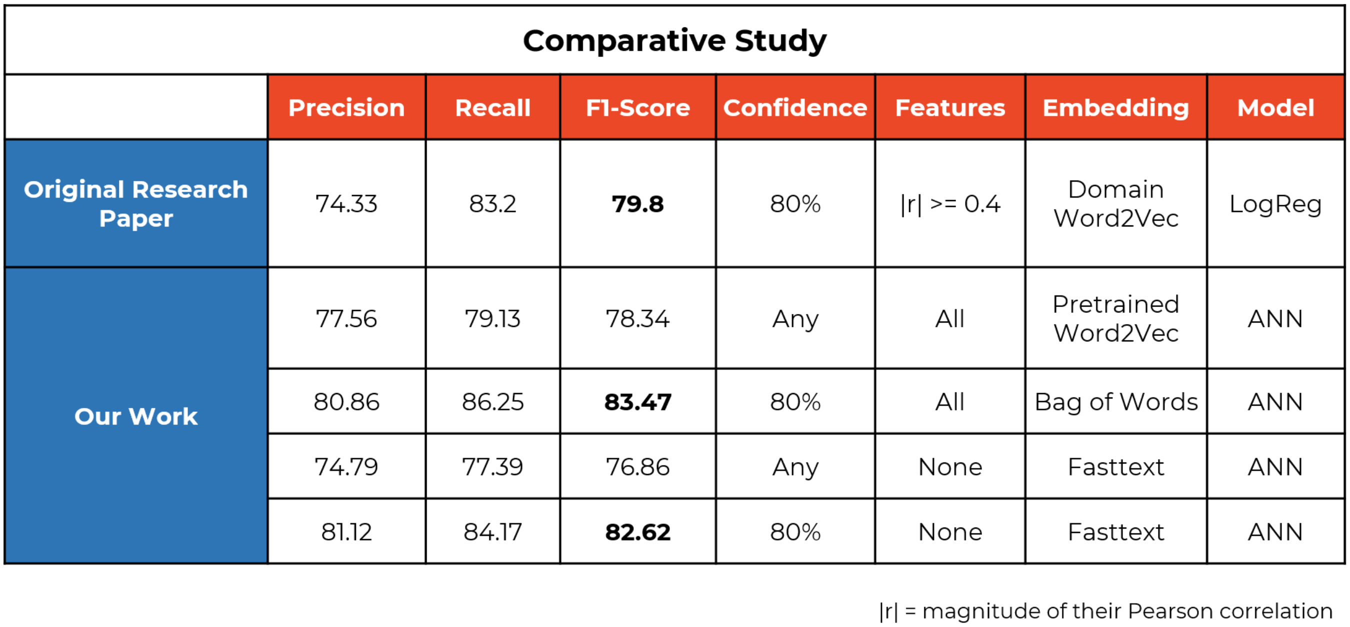


Figure 13: Word embeddings with features and 80% confidence

The following table shows the comparative study of the results only for the best outputs in different scenarios. Achieved better output by using advanced embedding techniques and neural networks.



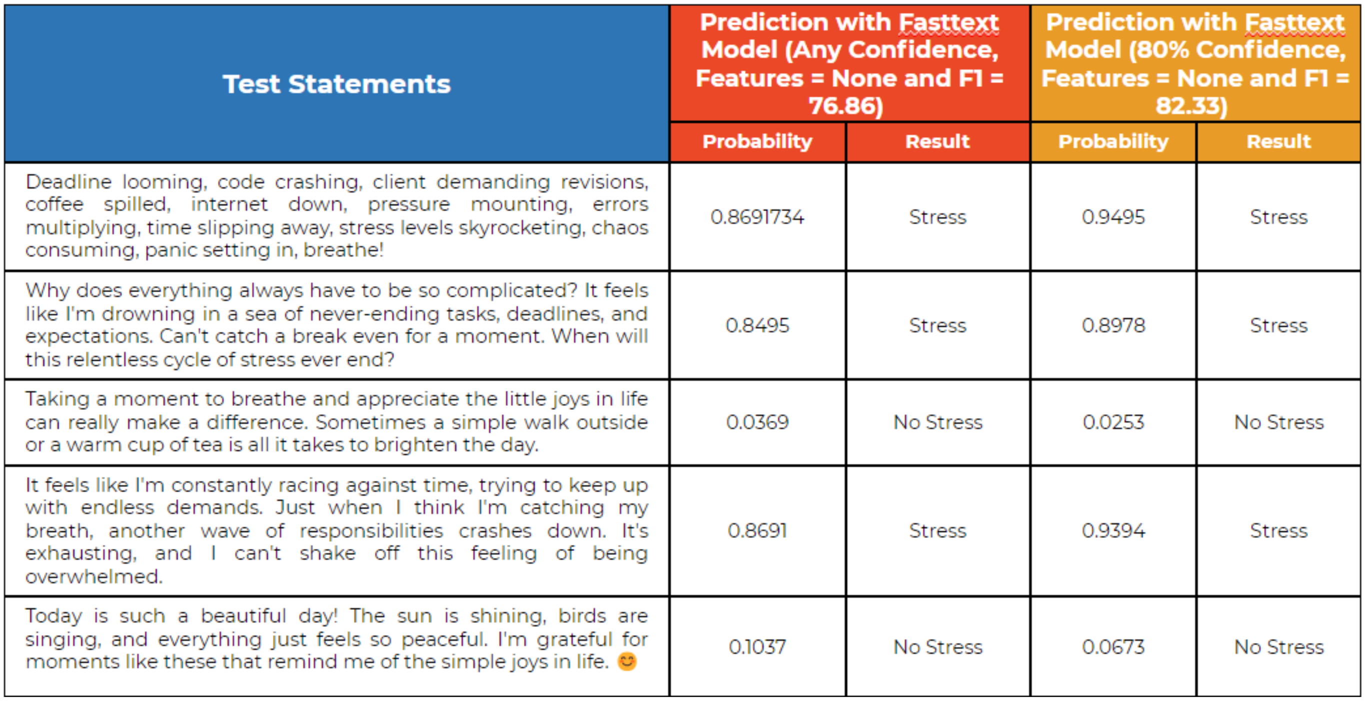
Comparative Study							
	Precision	Recall	F1-Score	Confidence	Features	Embedding	Model
Original Research Paper	74.33	83.2	**79.8**	80%	|r| >= 0.4	Domain Word2Vec	LogReg
Our Work	77.56	79.13	78.34	Any	All	Pretrained Word2Vec	ANN
	80.86	86.25	**83.47**	80%	All	Bag of Words	ANN
	74.79	77.39	76.86	Any	None	Fasttext	ANN
	81.12	84.17	**82.62**	80%	None	Fasttext	ANN

|r| = magnitude of their Pearson correlation

Table 4: Comparative study table of original vs our work results

4.9 Testing with Demo Data

Using multiple models, different embeddings plus different confidence levels we got multiple results. As shown in the previous table the best output we got using text and features combined is with Fast Text embedding using ANN. Here is a testing demo for different text and showing results which describe whether it is stress or not stress condition. The value shows that as it goes towards 1 it says stress condition. In case of not stress condition, with 80% confidence it is showing lesser value than the any confidence value, which makes it more evident that 80% confidence performs better in predicting the stress conditions. eg. In the no stress condition where model with any confidence making a probability of 0.0369, the model with 80% confidence giving a value of 0.0253 which shows better accuracy.



Test Statements	Prediction with Fasttext Model (Any Confidence, Features = None and F1 = 76.86)		Prediction with Fasttext Model (80% Confidence, Features = None and F1 = 82.33)	
	Probability	Result	Probability	Result
Deadline looming, code crashing, client demanding revisions, coffee spilled, internet down, pressure mounting, errors multiplying, time slipping away, stress levels skyrocketing, chaos consuming, panic setting in, breathe!	0.8691734	Stress	0.9495	Stress
Why does everything always have to be so complicated? It feels like I'm drowning in a sea of never-ending tasks, deadlines, and expectations. Can't catch a break even for a moment. When will this relentless cycle of stress ever end?	0.8495	Stress	0.8978	Stress
Taking a moment to breathe and appreciate the little joys in life can really make a difference. Sometimes a simple walk outside or a warm cup of tea is all it takes to brighten the day.	0.0369	No Stress	0.0253	No Stress
It feels like I'm constantly racing against time, trying to keep up with endless demands. Just when I think I'm catching my breath, another wave of responsibilities crashes down. It's exhausting, and I can't shake off this feeling of being overwhelmed.	0.8691	Stress	0.9394	Stress
Today is such a beautiful day! The sun is shining, birds are singing, and everything just feels so peaceful. I'm grateful for moments like these that remind me of the simple joys in life. 😊	0.1037	No Stress	0.0673	No Stress

Table 5: Test with Demo data table

4.10 Limitations

The study's limitations significantly impact the overall effectiveness of the models used for detecting mental stress in social media posts. Firstly, the relatively small dataset constrains the models' performance, leading to less robust and generalizable results. With a limited amount of training data, the models may not capture the full spectrum of stress expressions, affecting their accuracy and reliability. Secondly, the models encounter difficulties in identifying implicit stress, which is often subtly embedded within the text. Implicit stress indicators can be challenging to detect due to their nuanced and indirect nature, causing the models to miss these less obvious signs. Thirdly, there is a notable challenge in identifying stress that is focused on others rather than on the writer themselves. This limitation hampers the models' ability to generalize and accurately detect stress in different contexts, as stress expressions directed towards others may have different linguistic patterns. Lastly, the models do not fully account for the writer's framing and intention, which are essential for accurately interpreting the nuanced expressions of stress. Understanding the context and the writer's purpose behind the posts is crucial for a more precise identification of stress levels. These limitations highlight the need for more comprehensive datasets and advanced modeling techniques to improve the detection of mental stress in social media posts.

5. Conclusion

In conclusion, part one of the project aimed to develop an explainable intelligence-enabled model for predicting psychological disorders through interactive online questionnaires. However, we encountered limitations that prevented achieving this goal, including discrepancies in model accuracy and reduced performance due to Q-prioritization. Additionally, the Counterfactual Explanations Method (CEM) remains unimplemented. As a result, we could not achieve the desired accuracy or explainability.

In the second part we started working on a similar problem and here the paper focuses on detecting mental stress in social media posts by employing natural language processing (NLP) and machine learning (ML) or deep learning (DL) techniques. The research utilizes a variety of ML and DL algorithms on the Dreaddit dataset, placing particular emphasis on Fasttext embeddings. The findings reveal that artificial neural networks (ANN) combined with Fasttext embeddings are particularly effective in identifying signs of stress without needing additional features. Moreover, traditional methods like Bag-of-Words (BoW) and Word2Vec, when used in combination with embeddings and features, show superior performance. This demonstrates the potential of advanced techniques and underscores the need for further exploration into neural network-based models and pre-trained language models. The study also suggests that establishing benchmarks for Reddit datasets, similar to those for other social media platforms, could significantly aid in categorizing stress levels. Ultimately, the potential impact of this research lies in its promise to help mitigate mental health issues on social media by detecting signs of stress and providing necessary support.

6. Future Work

In future work, enhancing the model performance by leveraging unlabeled data will be a crucial step. By employing semi-supervised or unsupervised learning techniques, we can tap into vast amounts of unannotated social media content to improve stress detection accuracy. Furthermore, developing models that consider the writer's framing and intention will provide deeper insights into the context of the posts, allowing for a more nuanced understanding of stress indicators. Another important direction is to move beyond mere detection and delve into investigating the causes and effects of stress. Understanding the underlying factors and the impact of stress will enable more comprehensive mental health support mechanisms. Additionally, creating distant supervision techniques to expand the labeled data will be vital. These techniques can generate more annotated data from the unlabeled pool, thus enriching the training datasets and improving model robustness. By focusing on these areas, the project can significantly advance the field of mental stress detection on social media platforms, ultimately contributing to better mental health outcomes.

References

1. Alarcao SM, Fonseca MJ. Emotions recognition using EEG signals: a survey. *IEEE Trans Affect Comput*. (2017) 10:374–93. 10.1109/TAFFC.2017.2714671 [CrossRef] [Google Scholar]
2. Anjume S, Amandeep K, Aijaz A, Kulsum F. Performance analysis of machine learning techniques to predict mental health disorders in children. *Int J Innovat Res Comput Commun Eng*. (2017) 5. [Google Scholar]
3. Dabek F, Caban JJ. A neural network based model for predicting psychological conditions. In: *International conference on brain informatics and health*. London: Springer; (2015). p. 252–61. [Google Scholar]
4. Muhammad Abdul-Mageed and Lyle H. Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pages 718–728.
5. Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. In Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media
6. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
7. Yoon Kim. 2014. Convolutional neural networks for sentence classification. CoRR, abs/1408.5882.
8. Sharath Chandra Guntuku, Anneke Buffone, Kokil Jaidka, Johannes C. Eichstaedt, and Lyle H. Ungar. 2018. Understanding and measuring psychological stress using social media. CoRR, abs/1811.07430.
9. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.
10. Genta Indra Winata, Onno Pepijn Kampman, and Pascale Fung. 2018. Attention-based LSTM for psychological stress detection from spoken language using distant supervision. CoRR, abs/1805.12307