Ensemble Machine Learning Approach for Stress Detection in Social Media Texts

Maleeha Illahi[1], Isma Farah Siddiqui[1*], Qasim Ali[1], Fizza Abbas Alvi[2],

[1]Department of Software Engineering, MUET, Jamshoro, Pakistan
[2]Department of Computer Systems Engineering, QUEST, Nawabshah, Pakistan
[*]Corresponding author: isma.farah@faculty.muet.edu.pk

Abstract

Stress is an almost basic human instinct, especially in the internet age. People's health is being endangered by psychological stress. It is critical to diagnose stress early on to provide proactive therapy. In the age of the Internet, social media has an enormous impact on human thinking. It can cause mental health issues such as stress but can also play an important part in detecting it. Advancements in machine learning and natural language processing have enabled information extraction from a huge amount of raw textual data. In this study machine learning approach is used to detect stress in a social media text. Data used in this study is from Reddit. Classical and Ensemble machine learning approaches were used for stress detection. Classical Techniques include Decision Tree, Logistic Regression, support vector machine, Random Forest, and Naïve Bayes. Ensemble approaches used are boosting, bagging, and voting. The ensemble approach was able to provide better results than all machine learning baselines on the dataset. Also, it was able to outperform all non-transformed-based neural network architectures discussed in the baseline. The best-performing model in this study is Logistic regression with a 76.6% F1 Score.

Keywords—Machine Learning, Mental Health, Natural Language Processing, and Stress Detection in social media

1 Introduction

Social media has changed the way people interact with one another over the last decade. People actively share their daily activities, perspectives, feelings, viewpoints, hopes, aspirations, and emotions online, in addition to sharing fact-based information and news [1]. Despite the many benefits of social media, its use has a number of negative consequences such as Stress, depression, anxiety, addiction, cyberbullying, hacking, scams, and deceit. Images, text, and social media can all be used to detect stress. But primarily studies focused on textual data [2], [3]. Advances in machine learning are benefitting mental health treatment [14], [15], [16]. The Language-based signs are indications of bipolar disorder, depression, autism, personality disorder, and Schizophrenia [17]. This means that computational linguistics is important and can play a significant role in developing fresh insights about an individual's Emotions and mental health.

Natural language processing is a new technology that employs computers to comprehend human language. Many researchers attempted to improve the performance of stress detection by employing various Natural Language Processing (NLP) methods and text classification approaches. Machine learning methods have become significantly more famous for learning finer syntactic or semantic patterns in recent years.

In this study ensemble machine learning approach is used to perform stress detection from text. Both classical and ensemble machine learning approaches are created to perform stress detection. This study provides a comparison between state-of-the-art results with an ensemble learning approach and will discuss the benefits and disadvantages of using the ensemble technique. The following are the main contribution of this study:

1) This study applied ensemble machine-learning techniques for stress detection with higher accuracy.
2) Our model outperformed baseline models based

ISSN: 2523-0379 (Online), ISSN: 1605-8607 (Print)
DOI: https://doi.org/10.52584/QRJ.2002.15.

on Machine Learning and Nontranformer-based Deep Learning.
3) This study provides a novel and state-of-the-art comparison between Ensemble Machine Learning and state-of-the-art benchmark.

2 Literature Review

NLP is at the forefront of technologies to detect mental health issues such as stress and depression from social media texts. There are numerous studies that focus on the prediction of stress from text using NLP and machine learning in [4] researchers have used user-generated content from social media to predict stress using SVM and Naive Bayes but weren't able to achieve an accuracy of more than 63%. In [5] researchers have performed sentiment analysis for stress detection using the lexicon approach and evaluated models using Precision and recall. Models used were SVM and Naive Bayes so results are quite similar to [4] but improvement in data leads to a 4% increase in performance. In [6] researchers used mood detection from stress prediction and used SVM and KNN for modeling. Another approach proposed in [7] used depression detection with the help of a large number of tweets with Naive Bayes and SVM, this large number of tweets in datasets increased the performance of the model significantly and an f1 score of 83% is achieved. Another approach using a large dataset performed a comparison of machine learning and deep learning techniques to perform prediction sentiment [8]. To improve performance not only the quantity but quality of the dataset is also important. The dataset collected under clinical conditions performs better at this task as in [9] research used students to collect stress and later machine learning is used to perform future predictions and the model was able to achieve an F1 score of 85%. [10], who use Long Short-Term Memory Networks (LSTMs) to detect stress in a statement and Twitter data; [11], who investigate the Facebook and Twitter texts of users who scored high on an analytic stress survey; and [12], who detect stress on micro-blogging websites using a CNN (Convolutional Neural Network) and factor graph template with a suite of discrete features.

The focus of this study is to perform stress detection using text data only from social media content this requires a large standard dataset. [13] presents Dreaddit as a novel text corpus of extensive inter-domain social media text for stress detection. Our dataset includes 190,000 posts from five separate Reddit forums. They also performed the baseline benchmarking using different machine learning and deep learning techniques.

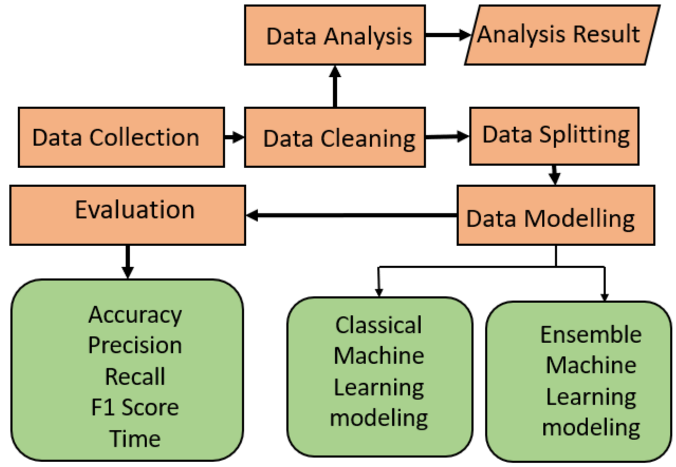

Fig. 1: Methodology Diagram for implementing Stress detection using machine learning

BERT outperformed all models in benchmarking. The majority of the studies were focused on either classical machine learning which is way too simplistic for this type of complex task, or deep learning-based language model but these models are huge they require a lot of computation to train and also need a very large amount of data one prominent technique to deal with these issues is ensemble machine learning which is the focus of this study. in this study ensemble of machine learning models is proposed and compared with the existing solutions.

3 Methodology

Figure 1 shows the flowchart diagram for implementing stress detection using classical and Ensemble approaches. Let's discuss each step of the methodology in detail.

3.1 Data Collection

The dataset is the most essential part of any machine learning problem. In this research, stress detection is performed using text data from social media. So, the required dataset must contain text and information regarding stress. This is a classification problem, so the stress column needs to be categorical. One way to obtain this type of data is to scrape social media and annotate the data which is a very difficult and tedious process so in this study, an existing dataset called Dreaddit [13] is used. This dataset contains 190,000 posts from Reddit and subreddit threads. It contains a column with text and a column with a stress label that shows if the text shows some stress on the writer of the text or not. Further analysis of the dataset is given in section 3 Data Analysis.

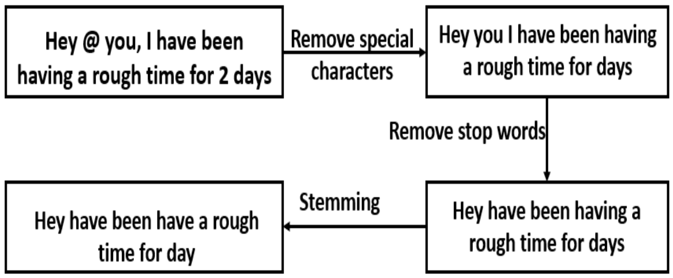

Fig. 2: Text Cleaning Example

3.2 Data Cleaning

Now we have data but to make it useful we need to clean the data. Specifically for this study as the data is in form of the text so text preprocessing approaches will be used to perform data cleaning. The first preprocessing step is to remove all the special characters from the text so that the end results of the text don't contain any character other than the alphabet. Once all the data is in the form of the alphabet now the next step is to perform stopword removal. Stopwords are words that are in large quantity but have very small syntactic values words such as "is, am and, are" these are words that are available in large amounts but contribute very little to the task of stress detection, so these texts are removed. The last preprocessing step is stemming. There are words in multiple forms such as the word try and trying both represents the same thing but due to their different form learning algorithm can consider different from ambiguity which is the reason that words are stemmed from their root word this preprocessing technique is called stemming. Figure 2 shows all the cleaning operations as discussed in this section.

3.3 Data Analysis

Data analysis is a way to under the data under consideration one of the key parameters to understand textual data are unigram, bigram, and trigram distribution is shown with their count in table 1. stress data. Stressed text instances are 1857 and non-stressed instances are 1696.

3.4 Data Splitting

In order to train a dataset using a machine learning algorithm two sets of the dataset are required, one for training and one for testing. For the given dataset researchers have provided the split where the training set contains 2838 instances and the testing set contains 715 instances. The training set is used in the modeling phase and the testing set is used in the evaluation phase. There are more instances with stress data than no stress.

3.5 Data Modeling

Once data is cleaned and split into the training set, the next step is performing machine learning modeling. For data modeling features are the building block of the model in this study two feature extraction techniques are used: Countvectorizer and TF-IDF. Countvectorizer makes it simple to use text information directly in ML and DL models like text categorization. CountVectorizer is a tool that converts a set of text documents into a vector of word counts. It also allows for text data pre-processing before producing the vector form. Term Frequency Inverse Document Frequency is abbreviated as TF-IDF. This is a method for counting the number of words in a collection of documents. In general, we assign a score to each word to indicate its relevance in the text and corpus. This approach is commonly utilized in information extraction and text mining. In this study 2 types of modeling techniques are used: classical ml techniques which include Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), SVM, Logistic Regression (LR), and KNN. Another modeling strategy is Ensemble which includes Boosting, Bagging, and Voting. Classical ML strategies are very straightforward so let's discuss ensemble machine learning techniques. Deep learning models especially transformer-based language model is very expensive to train and requires more computation for inferencing also data required to train these model is huge so in a use case such as this (Stress detection in text) data is very limited so one of the viable options is classical machine learning technique but then it has some performance restriction so to deal with ensemble machine learning models are used.

3.5.1 Boosting

Boosting is an ensemble modeling strategy that seeks to construct a strong classifier from a collection of weak classifiers. It is accomplished by developing a model in series utilizing weak models. First, a model is constructed using the training data. The 2nd model is then constructed in an attempt to address the faults in the previous model. This approach is repeated until either the whole training dataset is properly predicted or the max number of models is included. Boosting algorithms used in this study are gradient boosting, a boost, and histogram gradient boosting.

TABLE 1: Unigram, Bigram and Trigram count

Unigram	Count	Bigram	Count	Trigram	Count
like	1601	feel like	462	long story short	19
get	1458	tl dr	144	would greatly appreciate	16
feel	1454	panic attack	117	feel like go	15

3.5.2 Bagging

Bagging is an ML ensemble meta-algorithm meant to increase the reliability and accuracy of ML algorithms used in statistical regression and classification. It reduces variance and aids in avoiding overfitting. It is commonly used in decision tree approaches. Bagging is a subset of the model averaging method. Following are steps for Bagging Algorithms implementation:
Step 1: Multiple subsets with identical tuples are constructed from the actual dataset, choosing observations using replacement.
Step 2: For all the subsets from step 1 baseline model is created.
Step 3: Each of the models from step 2 is learned parallelly with all other models.
Step 4: Final prediction is obtained by combining results from all the models.

3.5.3 Voting

Voting is an Ensemble ML model that trains on a huge ensemble of models and performs prediction based on the highest likelihood of the chosen class being the outcome. It simply accumulates the results of each classifier that has been submitted through the Voting Classifier and determines the output label based on the largest majority of votes. Instead of developing separate specialized models and determining their correctness, we develop a single model that trains on these systems and estimates output relying on their aggregate majority of votes for each output label. Figure 5 shows the illustration of the voting process.

3.6 Evaluation

Once the modeling process is completed, the resultant models are evaluated to see how well they perform on unseen datasets. In this step test portion of the dataset will be used. The evaluation metrics used in this study are accuracy, precision, recall, and f1 score. These metrics are calculated with help of true positives, true negatives, false positives, and false negatives. In this use case, true positive is stressed text classified as stress, true negative non-stress data classified as non-stress, false positive means no stress classified as stress, and false negative is predicting stress as non-stress.

TABLE 2: F1 Score Classical and Ensemble

#	Name	FS	Classical	Bagging	Voting
1	SVM	CV	68.08%	73.49%	73.09%
2	LR	CV	73.04%	75.35%	73.04%
3	DT	CV	64.59%	67.11%	64.44%
4	RF	CV	66.85%	75.57%	75.37%
5	NB	CV	61.86%	63.15%	61.86%
6	KNN	CV	55.77%	47.39%	55.77%
7	SVM	TFIDF	68.08%	75.13%	75.03%
8	LR	TFIDF	76.67%	75.77%	76.67%
9	DT	TFIDF	62.48%	66.58%	62.34%
10	RF	TFIDF	67.98%	75.35%	74.51%
11	NB	TFIDF	60.58%	60.81%	60.58%
12	KNN	TFIDF	71.16%	71.99%	71.16%

These are represented by true-positive, true-negative false-positive, and false-negative. Equation 1, 2, 3, and 4 represents accuracy, precision, recall, and F1 Score.

4 Results

The implementation of experiments is done on CORE I5 machine with 8 GB of RAM. The programming Stack used is python with SKlearn, NLTK, pandas, numpy, and matplotlib. Python is used as a primary programming language for all experiments, NLTK is used for NLP operation, numpy and pandas for statistical analysis, and sklearn for Machine Learning modeling. In this study 2 types of ML approaches are used: classical and Ensemble and inside ensemble 3 subclasses of algorithms are used namely Bagging, voting, and boosting. For feature extraction Countvectorizer (CV) and TFIDF techniques are used; details of these techniques are given in the methodology. Results for classical, bagging, and voting are shown in table 2. The F1 score for all techniques for 3 techniques and 6 algorithms is shown in table 2. Table 2 also separates vectorization techniques such as CV and TFIDF. Based on the results the best-performing algorithm is Logistic Regression with the TF IDF technique. F1 score for TF IDF-based Logistic regression with classical, bagging, and voting is 76.67%, 75.77%, and 76.67% respectively. KNN with least is the least performing algorithm with a 47.39% f1 score with bagging. Logistic regression outperformed all because of its

TABLE 3: Boosting Result

#	Name	FS	A	P	R	F1_
1	AB	CV	68.4%	68.0%	73.2%	70.5%
2	HGB	CV	72.3%	71.2%	77.8%	74.4%
3	GB	CV	69.5%	69.2%	73.7%	71.4%
4	GB	TFIDF	67.7%	67.3%	72.9%	70.0%
5	AB	TFIDF	67.4%	67.8%	70.2%	69.0%
6	HGB	TFIDF	70.6%	69.7%	76.2%	72.8%

ability to perform well on binary classification tasks. Higher dimensionality of data made classification difficult for algorithms such as Naive Bayes and KNN. Table 3 shows the results for Boosting algorithms. Three types of Boosting algorithms are used: Adaboost (AB), Histogram Gradient Boosting (HGB), and Gradient Boosting (GB). All of these algorithms belong to the ensemble class of algorithms. All of the models for boosting techniques are also trained using both CV and TF IDF. CV-based HGB outperformed all models with a 74.4% f1 score. TF IDF-based AB is the least-performing model with an f1 score of 69%. The scope of the study is to compare classical and ensemble machine learning, so a comparison of results as shown in the table is made with the baseline results provided in [13]. Based on the results in table 4, models were able to outperform all the classical and non-transformer-based Deep learning models [13]. The best-performing model is TF IDF-based voting logistic regression and classical LR which has an f1 score of 76.67%, precision of 72.75%, and recall of 81.03%. Voting and classical results can match sometimes if the classical model is the best optimal model. In the case of this study, both voting and classical LR models are the best possible options. But it is not necessary that classical and voting results always match so it is always advantageous to check voting first voting and the classical approach.

Based on results from figure 3 for both CV and TF IDF SVM and LR were the best-performing models in terms of AUC ROC which validates the results from table 2. The least performing algorithms are KNN and NB which is also consistent with observations made from the earlier results.

5 Conclusion

In this study, 2 classes of machine learning approaches were used to detect stress in the text. The dataset used in this study extracted the text from Reddit. Six different machine-learning algorithms were used: SVM, LR, DT, RF, NB, and KNN. Two feature extraction

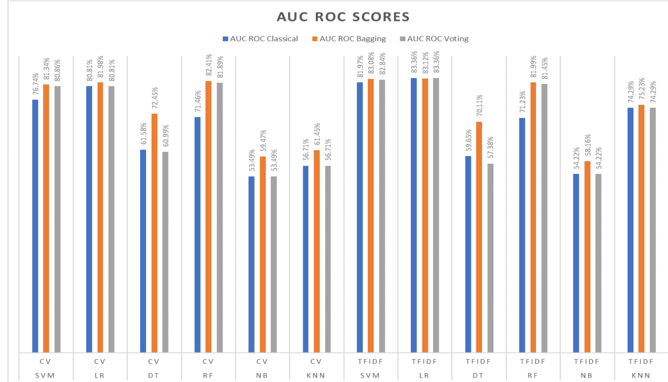

Fig. 3: AUC ROC Results

TABLE 4: Comparison with Baseline ML and Non-transformer-based DL techniques

Model	P	R	F1
LR-TFIDF-Voting/ Classical LR	72.75%	81.03%	76.67%
LR-TFIDF-Bagging	72.24%	79.67%	75.77%
LR-CV-Bagging	71.92%	79.13%	75.35%
n-gram baseline [13]	72.49%	76.42%	74.41%
GRNN w/ attention [13]	70.20%	77.24%	73.55%
LR-TFIDF-Bagging	71.61%	74.53%	73.04%

techniques were used CV and TF IDF. Also performed a comparison between the results obtained from the study to the baseline study. Based on the results the best-performing algorithm was Logistic regression the best-performing machine learning approach is voting and classical. For classical, bagging and voting TF IDF provided better results for Boosting CV was a more appropriate choice of feature selection. The limitation of the study is the scope of this study is limited to classical and Ensemble ML approaches, Deep learning, and the latest transformer-based architecture have not experimented in this study. So, for the future work of the study, deep learning BERT and classes of BERT will be applied and a new performance benchmark will be created for this dataset. The limitations of this study include a small dataset and a monolithic (Dataset obtained from a single source) which led to the problem of not being able to use deep learning very efficiently. In the future, this study will be extended to a more diverse and large amount of data so that a more generalized ML model can be created for stress detection and can be deployed in production.

References

[1] Murarka, Ankit, Balaji Radhakrishnan, and Sushma Ravichandran. "Detection and Classification of mental illnesses on social media using RoBERTa." arXiv preprint arXiv:2011.11226 (2020).

[2] Jadhav, Sachin, Apoorva Machale, Pooja Mharnur, Pratik Munot, and Shruti Math. ”Text based stress detection techniques analysis using social media.” In 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), pp. 1-5. IEEE, 2019.

[3] Can, Yekta Said, Niaz Chalabianloo, Deniz Ekiz, and Cem Ersoy. ”Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study.” Sensors 19, no. 8 (2019): 1849.

[4] Aldarwish, Maryam Mohammed, and Hafiz Farooq Ahmad. ”Predicting depression levels using social media posts.” In 2017 IEEE 13th international Symposium on Autonomous decentralized system (ISADS), pp. 277-280. IEEE, 2017.

[5] Baheti, R. R., and S. A. Kinariwala. ”Survey: sentiment stress identification using tensi/strength framework.” Int J Sci Res Eng Dev 2, no. 3 (2019): 1-8.

[6] Hussain, Jamil, Maqbool Ali, Hafiz Syed Muhammad Bilal, Muhammad Afzal, Hafiz Farooq Ahmad, Oresti Banos, and Sungyoung Lee. ”SNS based predictive model for depression.” In Inclusive Smart Cities and e-Health: 13th International Conference on Smart Homes and Health Telematics, ICOST 2015, Geneva, Switzerland, June 10-12, 2015, Proceedings 13, pp. 349-354. Springer International Publishing, 2015.

[7] Geissbühler, Antoine, Jacques Demongeot, Mounir Mokhtari, Bessam Abdulrazak, and Hamdi Aloulou, eds. Inclusive Smart Cities and E-Health: 13th International Conference on Smart Homes and Health Telematics, ICOST 2015, Geneva, Switzerland, June 10-12, 2015, Proceedings. Vol. 9102. Springer, 2015.

[8] Deshpande, Mandar, and Vignesh Rao. ”Depression detection using emotion artificial intelligence.” In 2017 international conference on intelligent sustainable systems (iciss), pp. 858-862. IEEE, 2017.

[9] Ahuja, Ravinder, and Alisha Banga. ”Mental stress detection in university students using machine learning algorithms.” Procedia Computer Science 152 (2019): 349-353.

[10] Winata, Genta Indra, Onno Pepijn Kampman, and Pascale Fung. ”Attention-based lstm for psychological stress detection from spoken language using distant supervision.” In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6204-6208. IEEE, 2018.

[11] Guntuku, Sharath Chandra, Anneke Buffone, Kokil Jaidka, Johannes C. Eichstaedt, and Lyle H. Ungar. ”Understanding and measuring psychological stress using social media.” In Proceedings of the international AAAI conference on web and social media, vol. 13, pp. 214-225. 2019.

[12] Lin, Huijie, Jia Jia, Jiezhong Qiu, Yongfeng Zhang, Guangyao Shen, Lexing Xie, Jie Tang, Ling Feng, and Tat-Seng Chua. ”Detecting stress based on social interactions in social networks.” IEEE Transactions on Knowledge and Data Engineering 29, no. 9 (2017): 1820-1833.

[13] Turcan, Elsbeth, and Kathleen McKeown. ”Dreaddit: A Reddit dataset for stress analysis in social media.” arXiv preprint arXiv:1911.00133 (2019).

[14] Luxton, David D. ”An introduction to artificial intelligence in behavioral and mental health care.” In Artificial intelligence in behavioral and mental health care, pp. 1-26. Academic Press, 2016.

[15] Kaur, Prableen, and Manik Sharma. ”Diagnosis of human psychological disorders using supervised learning and nature-inspired computing techniques: a meta-analysis.” Journal of medical systems 43 (2019): 1-30.

[16] Shatte, Adrian BR, Delyse M. Hutchinson, and Samantha J. Teague. ”Machine learning in mental health: a scoping review of methods and applications.” Psychological medicine 49, no. 9 (2019): 1426-1448.

[17] Cohen, Alex S., Kyle R. Mitchell, and Brita Elvevåg. ”What do we really know about blunted vocal affect and alogia? A meta-analysis of objective assessments.” Schizophrenia research 159, no. 2-3 (2014): 533-538.