COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

Classifying Mental Health Posts on Reddit: A Comparative Study of Tf-Idf and Word2Vec Embeddings

Exam paper

Natural Language Processing

[CDSCO1002U]

MSc Business Administration and Data Science

Group:

M4-NLP

Authors:

Mira Metzger (mime22af)

Marie Liljegren Gam (maga15ab) (102778)

Mathilde Lundsberg-Nielsen (malu22ae) (157498)

Michelle Judith Sara von Huth (mivo22ab) (158397)

Classifying Mental Health Posts on Reddit: A Comparative Study of Tf-Idf and Word2Vec Embeddings

Mira Metzger, Marie Liljegren Gam, Mathilde Lundsberg-Nielsen & Michelle Judith Sara von Huth

Copenhagen Business School, MSc. Business Administration and Data Science

Abstract

This paper investigates the application of various NLP techniques for the classification of mental health-related posts on Reddit. The primary focus is on the impact of tf-idf and Word2Vec embeddings on the performance of classification models of different complexities, specifically the Logistic Regression and Multi-Layer Perceptron. Unlike previous studies primarily aiming at detecting users with mental health conditions, our unique approach emphasizes the categorization of posts into appropriate mental health subreddits to support users to connect to the right communities. The study utilizes a dataset of over 700k posts from various subreddits related to mental health disorders. The report discusses the data pre-processing and model training to accurately classify users' post. Through this endeavour, we seek to improve user interactions, thus increasing the level of support within the Reddit community.

Keywords: Mental Health, Reddit, Classification, Logistic Regression, Multi-Layer Perceptron, Natural Language Processing, Word2Vec.

1 Introduction

Mental health is an increasingly pressing global concern, with rising rates of mental health disorders (WHO, 2022). Despite the availability of resources for support, access to mental health care is often limited or stigmatized, leaving many individuals without the necessary resources. The rise of social media has created new opportunities for individuals to share their personal experiences with mental health disorders. Reddit, in particular, has become a popular platform for users to discuss mental health related topics, and seek support from others. However, the sheer volume of content on the platform can make it difficult for users to navigate and find relevant information and communities to reach out to. The objective of this study is to address this issue by utilizing Natural Language Processing (NLP) techniques to classify posts into the appropriate reddit forums, called 'subreddits', based on their textual contents. By doing so, we can help users connect with others and receive the support they seek and need.

From a technical perspective, the specific aim of this study is related to an investigation into how different feature representations, namely tf-idf and Word2Vec, affects the accurateness of different classifiers. Two models, the discriminative Logistic Regression and a neural network, Multi-Layer Perceptron, were employed to address the task. In light of this, our study is devised to answer the following:

How does the use of tf-idf and Word2Vec embeddings impact the performance of classification models in the context of mental health related post classification on Reddit?

The primary analysis is based on an extensive dataset of posts from various subreddits related to mental health, including BPD, Anxiety, Depression, Mental Illness, Bipolar, and Schizophrenia. Through rigorous pre-processing and model

training, we demonstrate some of the remarkable advances of computers in learning and comprehending human language whilst also emphasizing its many inherent challenges and limitations.

2 Related Work

The growing tendency of addressing mental health difficulties in online forums such as Reddit is evident. Consequently, a number of research projects have utilized NLP and machine learning techniques to identify Reddit users who may be suffering from mental health concerns. A study that demonstrated substantial progress in this setting was published in 2017 by Yates et al. The scholars developed a novel method for detecting depression by employing a Convolutional Neural Network (CNN), a technique often used in image recognition tasks, to evaluate and classify text data from Reddit users. This unique CNN application marked an important advance toward the automatic recognition of mental health issues from social media messages.

Later in 2018, Cohan et al. extended automatic labelling to various mental health diseases and developed a new, sizable Self-reported Mental Health Diagnoses dataset from Reddit postings. The researchers took their work a step further by developing a binary classifier. This classifier was engineered using an array of classic machine learning techniques, including Convolutional Neural Networks, FastText, Logistic Regression, and Support Vector Machines (SVM). To enhance the efficacy of their Logistic Regression model, they utilized tf-idf and L2-normalized features for weighting. Additionally, the SVM model was trained with a focus on tf-idf bag-of-words features, thus exploring some of the possible impacts of different word embeddings for text classification.

In another significant contribution, Strube et al. (2019) explored the use of Hierarchical Attention Networks (HAN) for mental health detection. By effectively incorporating the concept of 'attention' in their model, they allowed their algorithm to focus on the more informative parts of a Reddit post, leading to better classification results. More recent studies have also utilized advanced techniques to improve performance. Dinu & Moldovan (2021) for instance, utilized the power of pretrained transformer models, such as BERT, XLNET, and RoBERTa to create a binary classifier that reflects the potential of these models in detecting depression.

In summary, these studies provide an in-depth look into the current state of research in the field, each advancing our understanding of how NLP and machine learning can be utilized to classify mental health conditions. While these studies have provided great insight, our approach takes a different perspective. Instead of trying to detect users with mental health issues, this study aims to support accurate automated organization of posts into the right mental health subreddits. In this way, users can find the right communities and support they need more easily.

From a technical standpoint, we intend to investigate how the use of tf-idf and Word2Vec embeddings effects the success of Logistic Regression and Multi-Layer Perceptron models in organizing mental health-related Reddit posts. To our knowledge, this approach has not been attempted before, thus our study will build on the efforts of the aforementioned scholars to offer an innovative solution to the issue.

3 Methodology

3.1 Dataset Description

This study utilizes a dataset published on Kaggle, that was created using the official Reddit API and PushShift API (Adha, 2022). The original dataset consists of roughly 702,000 rows of text data from 6 different sub-forums on Reddit. It contains five columns, including the post title (‘*title’*), contents (‘*selftext’*), publication time *(‘created_utc’),* whether the post is marked as appropriate for over-18’s only (‘*over_18’*), and finally the subreddit the post was submitted to (‘*subreddit’*).

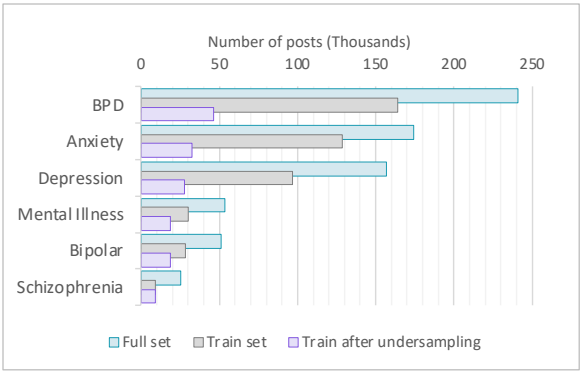

Figure 1: Class distribution before and after undersampling.

The '*title*' and '*selftext*' columns have 621,001 and 563,917 unique values respectively, with the difference being due to Reddit allowing posts that only contain a title and no other text. The 'subreddit' column has six possible values: BPD, Anxiety, Depression, Mental Illness, Bipolar, and Schizophrenia, defining the six target classes for the study. The number of posts in each class ranges from 25,365 (Schizophrenia) to 241,116 (BPD), as depicted by the blue bars in Figure 1, indicating a heavily class-imbalanced dataset.

A number of posts, around 96,600), had been deleted by either the original author or the subreddit's moderators, respectively indicated by '[deleted]' or '[removed]' in the '*selftext'* column. Additionally, likely due to data corruption or errors on loading, there were around 34,000 rows with null values in the '*selftext'* column. In total, this signifies under 19 pct. of the data being invalid, leaving just over 571,000 rows useful for the classification task. The pre-processing to resolve this is outlined in the sections below.

3.2 Training Strategy

To address the research problem, two distinct machine learning models were employed for the classification task: a Logistic Regression (LR), which is a relatively simple linear model that relies on a linear decision boundary to separate classes, and a Multi-Layer Perceptron (MLP), which is a more complex neural network consisting of multiple layers of neurons that can learn non-linear decision boundaries (Géron, 2022; Jurafsky & Martin, 2023). Furthermore, it was also decided that sparse and dense feature representations would be tested for both models, so that their performance could also be evaluated and compared. To create sparse vector representations, scikit-learn's CountVectorizer and TfidfTransformer were utilized (Pedregosa et al., 2011). To create dense vector embeddings, a Word2Vec model that has been pre-trained on part of the Google News dataset was used (RaRe Technologies, 2018). The dense and sparse vectors were each used to train both an LR and an MLP model, leading to four final models to compare, as seen in Figure 2.

The dataset was split into training and test sets using an 80:20 stratified sampling strategy to ensure that the class distributions in each set were representative of the distribution in the full set, as seen going from the blue to grey bars in Figure 1 (Géron, 2022). For the neural network approach, a subset of the training data was also reserved as a validation set, to be able to monitor the learning curves of the models during training.

The LR model was implemented using Scikit-learn, whilst the MLP was constructed using TensorFlow Keras. A combination of cross-validated grid search and 'babysitting' was employed for the hyperparameter tuning process of the LR model (Yang & Shami, 2020; Géron, 2022). Where the former allows for systematic exploring of a set combination of parameters, the latter enables manual monitoring of the grid search parameter space based on optimal configurations found in previous iterations, ultimately permitting the exploration of numerous combinations and their impacts on performance. For the MLP, simple manual configuration, i.e., 'babysitting', was performed to preserve control.

3.3 Data Pre-Processing

Figure 2 shows the general workflow followed for the pre-processing and splitting of the data. Initially, rows containing null values or the

designators for a removed or deleted post were dropped. Then, the few posts that contained references to other deleted or removed posts were also dropped, as this meant there was context and information missing from the text. Then the columns were adjusted to the correct format for later use. This involved amending the subreddit names for consistency and concatenating the *'title'* and *'selftext'* columns into a new *'post'* column, allowing both to be used for classification. This resulted in a dataset of around 570,000 rows in two columns, spanning 6 classes.

Once the dataset had been split, as previously described, the class imbalance could be addressed on the training set only, to avoid data leakage between the train and test sets (Ma & He, 2013). It is furthermore important to not modify the test set as, when predicting on unseen data, the class distribution of this data should reflect the real-world distribution.

The original dataset had an uneven class distribution: for every 1 Schizophrenia post there were roughly 10 BPD posts, 7 for Anxiety, 6 for Depression, and 2 posts for each of the Mental Illness and Bipolar subreddits (a ratio of 10:7:6:2:2:1). This indicates a substantial class imbalance. To mitigate the risk of the models skewing in favour of the majority classes, it was decided that undersampling the training set would be appropriate (Ma & He, 2013). Due to the severity of the imbalance, if a fully equally balanced approach was taken, the training set would be reduced to only 55,830 rows out of the possible 457,000 – just over 12%. Therefore, a more conservative approach was taken with a target distribution of 5:3.5:3:2:2:1, resulting in a train set of around 153,500 rows. Iterative experimentation was utilized to find this ratio at which the classifiers began to be able to distinguish the classes to a satisfactory level (Ma & He, 2013). The imbalanced learning library Imblearn's RandomUnderSampler was used to achieve the target ratio, and the new undersampled distribution of the train set is seen in the purple bars of Figure 1 (Lemaître et al., 2017).

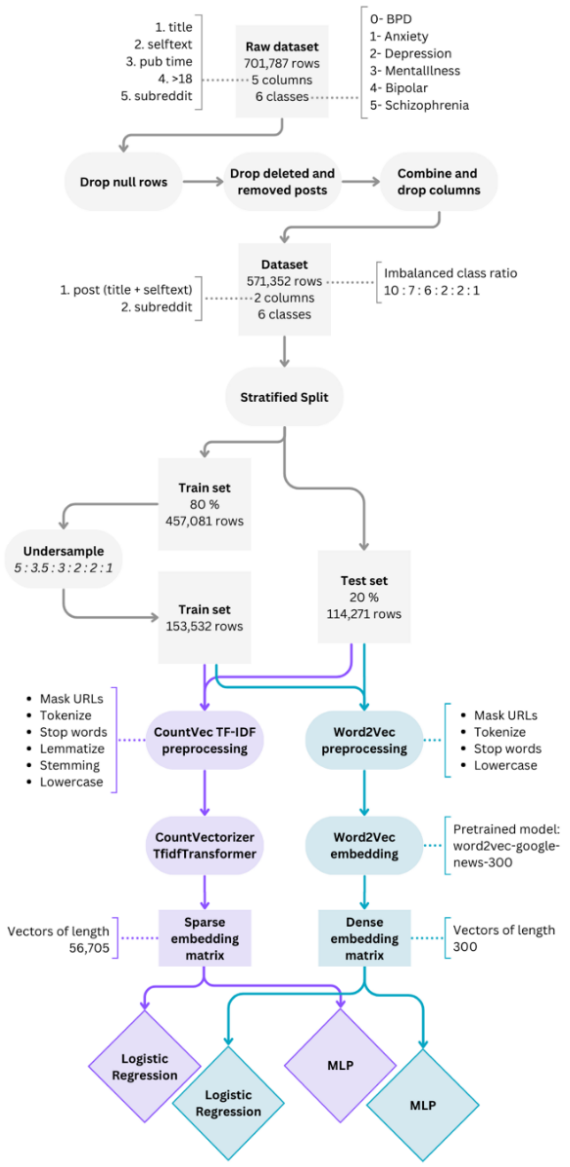

Figure 2: Data processing and training flow.

The train and test sets were processed individually, but following the same method, to avoid data leakage between the two (Pedregosa et al., 2011). As indicated by purple in Figure 2, the pre-processing before creating sparse embeddings with tf-idf included masking URLs, tokenization, removing stop words, lemmatization, and stemming. Scikit-learn's CountVectorizer and TfidfTransformer were then used to fit and transform the tokens into sparse vector representations of each post, where each vector is of length 56,705 – equivalent to the size of the corpus vocabulary

(Pedregosa et al., 2011). Conversely, as indicated by blue in Figure 2, only masking URLs, tokenization and removing stop words was performed for the dense vector pre-processing, because the pre-trained word2vec model used was not trained on lemmatized texts, and as lemmatizing and stemming may in fact remove context. The pre-trained Word2Vec model *word2vec-google-news-300* was used to transform the tokens into dense vector representations of each post, where each vector has length 300, corresponding to the vector dimensionality set when the pre-training was performed (RaRe Technologies, 2018).

3.4 Tf-Idf & Word2Vec

After pre-processing the data, the next step involved extracting features from the text which will serve as an input for the LR and MLP models. To generate the features, both tf-idf and w2v models are widely used (Cahyani and Patasik, 2021). Previous research conducted by Cahyani and Patasik (2021) on emotion classification concluded that tf-idf outperformed w2v in terms of performance metrics for a SVM model. The aim of this project is to determine if these results are replicable for the MLP and LR models for classifying reddit posts, and in what way the results differ between the two chosen models and feature representations respectively. Identifying the optimal text representation approach is crucial as it can have a large impact on the performance of the models and computation time (Cahyani and Patasik, 2021).

Tf-idf is a feature extraction technique that generates high-dimensional sparse matrices containing mostly zeros (Jurafsky & Martin, 2023). Tf-idf assigns weights to words in a document relative to their importance (Cahyani and Patasik, 2021). To evaluate the importance of a word, tf-idf both considers how many times a word appears in a document (tf), and its frequency across all documents (idf) (Cahyani and Patasik, 2021). This means that the weight of a word is higher when the frequency in one document is high, but low in all other documents. The weight assigned to that word then underscores its importance in distinguishing between the documents. A drawback from this approach is that the vector representations do not capture context or where words have similar meanings (Jurafsky & Martin, 2023).

The w2v is a deep learning technique that generates word embeddings from unlabelled text data (Jiang et al. 2020). Word embeddings are numerical dense vectors that represent each word in a high-dimensional space. The idea is, that similar words appear in the same context, and are thus represented by similar vectors. W2v is therefore capable of understanding the semantics of words. One benefit of this approach is that even if rare words that were not present in the training data appear in a new Reddit post, the model can still understand their meaning and thus make a correct prediction (Jurafsky & Martin, 2023). Additionally, generating word embeddings with w2v is fast and efficient and allows making use of pre-trained word embeddings (Jurafsky & Martin, 2023).

One of the advantages of utilizing pre-trained word embedding models is that they eliminate the need for training data to learn the word representations (Jurafsky & Martin, 2023). For this reason, the pre-trained model *word2vec-google-news-300* was used that was trained on a Google News dataset containing around 100 billion words (RaRe Technologies, 2018). The model consists of 300-dimensional vectors with 3 million words and phrases. The language used in news articles substantially differ from the language used on platforms such as Reddit, which may introduce some challenges and potential issues. For example, slang or specific mental-health related words and abbreviations might not be part of the Google news dataset and therefore not captured by the word embedding model. Despite these challenges, using a pre-trained model is preferred over a custom embedding model due to the limited training data and computational resources available in this study.

3.5 Modelling Framework

In the following subsections, the two model types implemented to solve the classification problem will be outlined. Following this, their individual hyperparameter tuning processes and optimal parameters utilized in the final models will be described. The performance of the models will later be studied in parallel, with the primary objective of evaluating and comparing which of the two is better suited for modelling the data at hand, combined with a cross-evaluation of the most appropriate method for feature extraction as outlined in section 3.4.

3.5.1 Multi-Layer Perceptron (MLP)

To classify Reddit posts concerning mental health into appropriate subreddits, the initial classification models will be two MLP implementations; one for each feature extraction type. Compared to the Single-Layer Perceptron, the Multi-Layer Perceptron can execute complex tasks such as regression and classification into multiple groups (Géron, 2022). The MLP architecture is composed of a single passthrough input layer, then a single or several hidden layers, and one final output layer. All layers are fully connected and include a bias neuron.

The MLP relies on backpropagation as optimization algorithm (Géron, 2022). Backpropagation is a method commonly used in neural networks to adjust the weights and biases of a network, based on the error between the predicted and the true output (Géron, 2022). This is done by computing the gradient of the error with respect to the weights and biases at each layer of the network and adjusting them in such a way that it shrinks the error. The adjustments are made by propagating the error back through the network layers, starting at the output layer, and moving in the direction towards the input layer (Géron, 2022). In this way, the backpropagation algorithm makes the network capable of understanding how each connection weight and bias term needs to be tweaked to be able to continuously reduce the error. After the network has learned the gradients, it simply performs a regular gradient descent step and repeats until it converges to the solution.

The hyperparameter tuning process was undertaken through manual 'babysitting' for both the w2v and the tf-idf MLPs (Yang & Shami, 2020). This was done to maintain control over the training flows, and to closely monitor the effects of adjusting the relevant hyperparameters as also outlined in section 3.2 (Géron, 2022). The optimal architecture for the w2v model in this setting, comprises three hidden layers with 64, 128, and 256 neurons respectively. The implicit input layer in the first dense layer were specified with 300 neurons to match the length of the pretrained w2v embeddings. All dense layers rely on the Rectified Linear Unit (ReLU) activation function and are followed by a batch normalization layer to normalize the inputs of the succeeding layer. To combat overfitting, dropout regularization with a rate of .5 was also applied following every layer.

On the other hand, the optimal performance of the tf-idf model was reached with three hidden layers with 50, 100, and 150 neurons respectively. The implicit input layer was in this case specified with 56,705 neurons, corresponding to the length of the vectors and the corpus vocabulary. As with the w2v model, all layers rely on ReLU activation, and are followed by a batch normalization layer to normalize inputs iteratively between layers (Géron, 2022). Dropout regularization was also specified for the tf-idf implementation, however, a rate of .2 was found to be more optimal here. As discussed in section 3.4, compared to the dense representations of the w2v embeddings, the tf-idf representations are high dimensional and sparse, where many features have zero values (Jurafsky & Martin, 2023). Thus, L2 regularization was applied to the input layer to help control the magnitude of the weights and encourage the model to assign smaller weights to less informative or noisy features (Géron, 2022).

As the MLPs are implemented for multi-class classification, the output neurons need to correspond to the six subreddit labels respectively

(Géron, 2022). Thus, the output layer of both models was specified to have 6 neurons and rely on the softmax activation function to produce a probability distribution over the classes. Both models were additionally trained to minimize the categorical cross-entropy loss function by means of the Adam optimizer (Géron, 2022).

During training, each model processed a batch of 32 samples at once before adjusting weights and biases (Géron, 2022). The number of epochs i.e., the number of times the dataset would be evaluated by the models was specified at 100. However, early stopping with a patience of 5 was also specified leading to 14 evaluated epochs for the w2v MLP, and 41 evaluated epochs for the tf-idf MLP. At the end of each epoch, the performance and loss for both models were evaluated on the validation subsamples to test their generalization abilities iteratively and monitor developments concerning over- and underfitting.

3.5.2 Logistic Regression (LR)

While the MLP is a powerful and versatile algorithm, the LR is a simple yet effective method that is commonly used in binary classification problems but can be extended to handle multi-class classification tasks as well (Jurafsky & Martin, 2009). The LR model estimates the probability of each possible class given the input features, which in this case is the extracted vector representations (Géron, 2022). It uses a linear combination of the input features and applies the softmax activation function to transform the output into a probability distribution over the six possible classes, making it particularly useful for multi-class classification problems like the one undertaken in this study. (Vimal & Kumar, 2020).

To optimize the hyperparameters of the LR classifier, we employed a cross-validated grid search approach combined with the nicknamed 'babysitting' method as outlined in 3.2 Training Strategy (Yang & Shami, 2020). This method allowed us to maintain better control over the training process, and closely observe the effects of tuning adjustments on the relevant parameters. The hyperparameter combination leading to the best performance were: *'C'*: 10, *'max_iter'*: 100, *'penalty'*: L2, *'solver'*: Saga, *'tol'*: .001.

'C' is the inverse of regularization strength; therefore, a smaller *'C'* gives greater regularization to minimize overfitting (Vimal & Kumar, 2020). *'Max_iter'* denotes the maximum number of iterations for the solver to converge, with the algorithm ending after this number of steps if an optimal solution is not discovered. The *'penalty'* parameter sets the type of regularization, with L2 applying a penalty equal to the square of the size of the coefficients. The *'solver'* parameter specifies the algorithm used in the optimization problem, with Saga being ideal for large datasets. Finally, *'tol'* specifies the stopping criterion tolerance, terminating iterations when the difference in loss function values between two iterations falls below this point.

LR is an appropriate algorithm for this Reddit post analysis because it provides easily interpretable results. The model calculates the importance of each input feature and assigns a corresponding coefficient value, which provides information and understanding as to the impact of each feature on the predicted outcome (Vimal & Kumar, 2020). This means it can easily identify important words or phrases in the text that are associated with specific subreddits.

Another advantage of LR is that it is computationally efficient and relatively fast, making it a great choice for analysing large datasets with many input features (Jurafsky & Martin, 2009; Vimal & Kumar, 2020). Additionally, the LR model has a low risk of overfitting, which is important because we want to avoid fitting to the noise in the data instead of the underlying patterns (Vimal & Kumar, 2020).

3.6 Evaluation Metrics

To gain comprehensive insights on the respective models' performance in the classification pro-

blem, a range of evaluation metrics will be examined after making predictions on the test set. Due to the relative imbalanced and multi-class nature of the task, the general accuracy of the models is discarded as an overall performance measure. The primary analysis will instead be based on measures of precision, recall and F1-scores for each of the respective classes and the weighted average F1 for an overall assessment.

Precision measures the fraction of correctly predicted positive instances, out of the total number of instances predicted as positive (Géron, 2022). High precision thus signifies a low frequency of false positives. Conversely, recall denotes the proportion of correctly predicted positive instances, relative to all of the actual positive instances, thus, a high recall translates to a low rate of false negatives (Géron, 2022).

In the present setting of developing an algorithm to support smoother access to relevant information and communities on Reddit, the cost of false negatives can be argued to be higher than the cost of false positives. Predicting a post to be relevant to a given subreddit, without it being so (false positive), can lead to irrelevant posts being displayed in a subreddit, but primarily compromises the user experience through cluttering if the frequency of these instances is high. False negatives on the other hand, when the model fails to identify a post as relevant to a particular subreddit when it in fact is, can result in valuable content being overlooked, and not reaching its intended audience, and is thus of higher cost to the individual user.

As precision and recall exhibits a trade-off - if one increases, the other one tends to decrease – finding the optimal decision threshold is a balancing act, and thus it is not valuable here to completely disregard one over the other, even if one appears more important (Géron, 2022). Thus, the harmonic mean of the two, the F1-score, becomes a valuable metric for class-individual performance assessment. As previously stated, the classes in the undersampled dataset are still not uniformly distributed. For this reason, the weighted average F1-score will be the primary measure to evaluate the models' overall performances, as it considers the contribution of each class based on its support measure, which due to the irregular ratios, naturally will differ between classes (Géron, 2022).

4 Results

In this section, the effectiveness of the models for sorting mental health-related Reddit posts into relevant subreddits using both LR and MLP algorithms will be analysed. This analysis will be conducted for each of the two feature representation approaches: w2v and tf-idf. The training of the models will subsequently be evaluated to assess if under- or overfitting is of concern. Finally, we will compare the complexity and running time of each model using each of the two feature extraction approaches.

The LR models were trained using the two separate feature extraction methods, w2v and tf-idf. The performance on the test set using w2v embeddings, as measured by the confusion matrix and weighted average F1-score, was .70. As seen in Table 1, the precision, recall, and F1-score for each class varied greatly, with the BPD and Anxiety classes having the highest F1-scores of .77 and .79, respectively. Looking at the confusion matrix (Appendix A), out of the actual 41,025 posts belonging to the BPD class, 32,693 posts were accurately predicted by the model, as were 24,534 for the Anxiety class out of 32,113 posts. However, the model struggled for the Mental Illness and Schizophrenia classes, showing lower F1 success rates of .30 and .42, respectively, and could only predict 1,961 posts to the Mental Illness class out of 7,483, and 987 posts for the Schizophrenia class out of 2,323 posts.

The LR model with tf-idf feature extraction demonstrated a significantly better performance over the w2v implementation, with a weighted average F1-score of .77 – a 7 percentage-point improvement compared to the w2v model. The precision, recall, and F1-scores also varied between

Table 1: Results presented as performance metrics.

Logistic Regression - Word2Vec ($F1_{weighted\ avg.}$ = .70)

Subreddit	Precision	Recall	F1-score	Support
BPD	.74	.80	.77	41,025
Anxiety	.82	.76	.79	32,113
Bipolar	.55	.56	.56	7,118
Depression	.65	.66	.66	24,194
Mental illness	.34	.26	.30	7,483
Schizophrenia	.42	.42	.42	2,323

Logistic Regression - Tf-Idf ($F1_{weighted\ avg.}$ = .77)

Subreddit	Precision	Recall	F1-score	Support
BPD	.83	.83	.83	41,028
Anxiety	.86	.83	.85	32,114
Bipolar	.69	.68	.68	7,118
Depression	.71	.74	.73	24,198
Mental illness	.40	.40	.40	7,487
Schizophrenia	.56	.59	.58	2,326

MLP - Word2Vec ($F1_{weighted\ avg.}$ = .69)

Subreddit	Precision	Recall	F1-score	Support
BPD	.78	.76	.77	41,025
Anxiety	.84	.73	.78	32,113
Bipolar	.44	.65	.53	7,118
Depression	.66	.65	.65	24,194
Mental illness	.31	.29	.30	7,483
Schizophrenia	.29	.56	.38	2,323

MLP - Tf-Idf ($F1_{weighted\ avg.}$ = .65)

Subreddit	Precision	Recall	F1-score	Support
BPD	.73	.73	.73	41,028
Anxiety	.77	.70	.74	32,114
Bipolar	.43	.58	.50	7,118
Depression	.58	.64	.61	24,198
Mental illness	.30	.19	.23	7,487
Schizophrenia	.34	.45	.39	2,326

classes here, although the BPD and Anxiety classes had the highest F1-scores, similar to the w2v model. The F1-score for the BPD class was .83, whereas it was .85 for the Anxiety class. Overall, the tf-idf and LR model combination performed moderately in categorizing posts into relevant subreddits, with F1-scores ranging from .40 to .85. Furthermore, when examining the details of each class's row in the confusion matrix on the test data (Appendix A), the BPD class also naturally had the highest number of correctly predicted posts, at 34,079 out of 41,028 posts, just barely beating out the w2v implementation. The model continued to have the most difficulty with the Schizophrenia and Mental Illness classes, despite improvements in performance. For the Mental Illness class, out of the actual 7,487 posts, the model accurately predicted only 2,983 posts. With exception of the Mental Illness class, a general tendency for both LR models is evident in that the performance for the individual classes is relative to the number of instances in the class, as reflected by the support measures and F1-scores: the more instances in a class, the better the performance.

Assessing the performance of the two MLP classifiers from an overall perspective, neither of the weighted average F1-scores managed to surpass those of the LRs. The highest performance in the neural network approach was measured at .69 for the w2v implementation, which can possibly be reasoned by the semantic representations learned by these embeddings having supported the model in capturing more nuanced relationships in the data (Jurafsky & Martin, 2009).

Delving into the classes' individual performance, Table 1 reveals that the w2v MLP has only marginally lower F1-scores for Anxiety, Bipolar, Depression, and Schizophrenia; its performance on the remaining classes is equal to the LR w2v model's. Thus, a pattern of higher support corresponding to better performance is also present here, also with exception of the collective Mental Illness class. As can also be seen in the results for the w2v LR model, we see specifically low or moderate performance for the all-encompassing Mental Illness class but also for Schizophrenia, presumably reasoned by overlapping post content in the former, and the lower number of instances belonging to the latter. With reference to the higher cost of false negatives for the given setting, the MLP however outperformed the LR in recall for the Schizophrenia class, with a .56 measure compared to .42 for LR, also for the Bipolar class.

In both cases, the trade-off between precision and recall are evident, as the bettered recall has negatively impacted the precision for these labels, leading to overall worse F1s for the MLP across these classes as well.

The losing match between feature representation and model is seen for the MLP with tf-idf extractions, demonstrating the lowest weighted F1 of all combinations at .65. The high dimensions of the sparse tf-idf representations were thus better captured by the simpler approach, presumably also due to their linear nature. None of the classes for the tf-idf MLP outperformed any other measures across all models and classes, however, the pattern of higher support leading to higher performance was yet reconfirmed.

On a closing note, we see that for the discriminative approach, the sparse embeddings lead to better performance in contrast to the dense w2v embeddings prevailing for the neural network. This outcome can be reasoned by several factors. For one part, the LR assumes a linear relationship between features and target variables. As the tf-idf embeddings can capture linear patterns as well, it can be deemed natural that the logistic regression performs better with this construction (Jurafsky & Martin, 2009). On the other hand, the nuanced features provided by the w2v embeddings successfully allowed the MLP to explore non-linear patterns in the data as it is knowingly capable of.

Confusion matrices for all four models can be found in Appendix A.

4.1 Training Evaluation

To comprehensively evaluate the learning procedure and monitor issues of under- and overfitting for the MLPs, training and validation loss curves were plotted over epochs during training (Géron, 2022). For the tf-idf implementation we see that both training and validation loss reduces at a high rate within the first 2 epochs, followed by a period of relative stability thereafter. The close proximity of the two signifies that the model is learning and generalizing well for the unseen validation subsets (Géron, 2022). The initial drop in loss suggests that the model quickly picks up on the patterns of the training data, and that the adjustments of weights and biases based on the error signal and the gradient of the loss function helps the model converge at an optimum relatively quickly (Géron, 2022). A recurrent pattern in the training process was that no matter the configuration of the model, the loss would always stabilize around approximately 1-1.2. This may pertain to the inherent complexity of the data, stemming

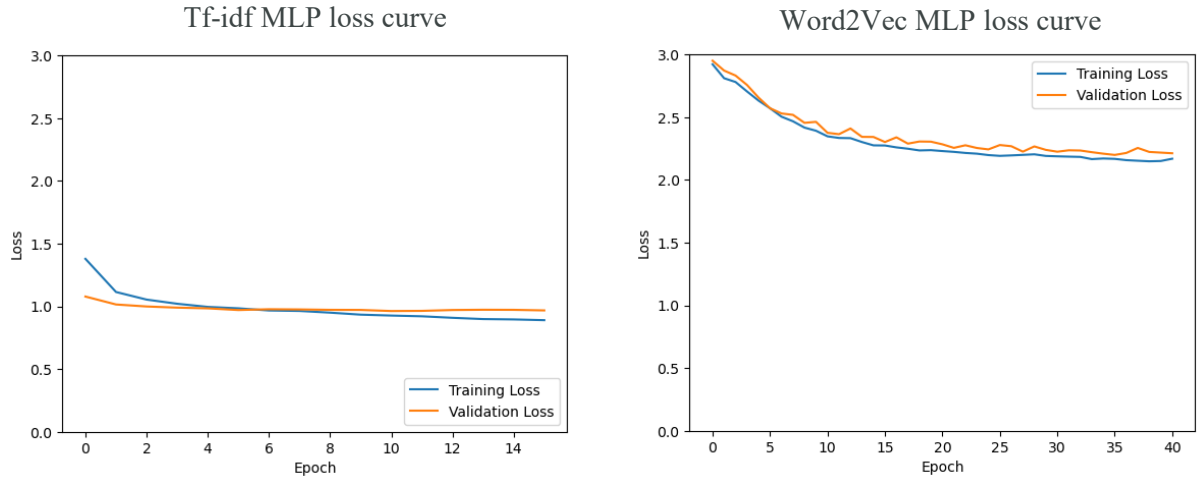

Figure 3: MLP learning curves (left: tf-idf, right: word2vec).

from its 'slangy' every-day tone, and thus nature, or the model possibly being either insufficiently or excessively complex to capture these inherent patterns more accurately.

The w2v MLP also made gradual downward progress in minimizing the loss function, however, the average loss of this model is higher, and the validation loss in particular shows smaller fluctuations over epochs, compared to more stable curves for the tf-idf model. The higher loss observed for this model may pertain to the model's ability to capture subtle nuances and semantic relationships in the input data, given the w2v embeddings – abilities that the tf-idf representations do not reflect (Jurafsky & Martin, 2009). The added complex relationships represented by the w2v embeddings are generally more expressive and as demonstrated, the optimization of the model required more training epochs to learn and utilize the information encoded in them. Despite a lowered rate of learning, we also see close proximity of training and validation loss indicating overall relatively good discriminative power for the w2v model (Géron, 2022). Also, the loss reached its minimum at a rather high level across different configurations, thus re-emphasising the complexity of the general task.

For both LR models, cross-validation scores were monitored and found to remain consistent over the validation folds without heavy fluctuations, which is a contraindication of overfitting for both models (Géron, 2022).

4.2 Complexity & Running Time

In the following paragraphs, the complexity and running time of the LR and MLP models are examined by comparing them with each other, as well as when utilizing the w2v and tf-idf feature extraction techniques, respectively.

Following the structure of the runtimes outlined in Table 2, the w2v LR model required a total CPU time of 5 minutes and 28 seconds. Just minutely faster was the tf-idf LR model, which took roughly 3 minutes and 16 seconds to run. Thus, a reduction in runtime was witness from dense to sparse representations for the linear model. Relating complexity to performance, the tf-idf LR model outperformed its w2v counterpart, with both a faster running time and a higher F1-score.

Table 2: CPU running times.

Model	CPU running time
LR - Word2Vec	5 min 28 sec
LR - Tfidf	3 min 16 sec
MLP - Word2Vec	4 min 9 sec
MLP - Tfidf	12 h 52 min 24 sec

Evaluating the efficiency of the MLPs we see very divergent runtimes, in each end of the spectrum. The w2v model had the overall quickest CPU runtime of roughly 4 minutes, whereas the tf-idf model required almost 13 hours. Thus, the tf-idf MLP not only demonstrated the poorest performance measures but can also be deemed the least efficient of all combinations.

In addition to an assessment of the specific CPU times of the models, attention should be paid to the runtime of the feature extraction techniques themselves as well. Given the large and sparse nature of the tf-idf extractions, the vectorization process had a natural longer execution time of 18 minutes and 31 seconds compared to the dense w2v representations, running in simply 23 seconds. Whether the decreased efficiency merits the gains in performance will however depend on the implementation objective and setting.

5 Discussion

The subsequent discussion is structured in to four separate sections: 5.1 Model Diagnostics, 5.2 Error Analysis, 5.3 Ambiguity & Practical Implications, and 5.4 Limitations. Whereas the model diagnostics outlines a general comparison of models, the error analysis and thus ambiguity &

practical implications sections will be based on the predictions of the optimal model and feature representation configuration only.

5.1 Model Diagnostics

Observing the results of the four models in parallel, it is evident that the simple combination of the linear LR model with the sparse tf-idf representations outperformed all other combinations in terms of performance as well as computational efficiency. Thus, the added complexity of the neural network was not justified for the given setting. Despite its complexity, the MLP model does not always deliver superior performance, and the increased processing resources and time required may not always convert to improved results. In fact, as demonstrated in the present study, the simpler model can sometimes outperform the complex one, and such case translates to a profitable situation where the task at hand benefits from both a decreased complexity, but also computational efficiency.

At the same time, the findings support the notion that the semantic and contextual information captured by the w2v embeddings was not essential for accomplishing the task either. This can be reasoned by several factors. For one part, it is imaginable, that due to the short and concise language of such posts, the amount of contextual information will be limited, hence, restricting the effectiveness of w2v. Conversely, it can be argued that specific keywords related to symptoms or specific emotions have high likelihood of reappearing within the studied subreddits, making them prominent features for the classification that can thus be easily and better captured using the tf-idf approach.

To address the proposed research question based on the points above, it can be concluded that the results of our study indicate that the choice of feature representation for a classification task, in the field of NLP, can have a notable impact on performance. Specific attention should hence be paid to the linguistic properties of the data and the choice of model/approach, and several techniques and combinations should be explored and assessed to be able to make decisions about the optimal choice for a given problem. The obvious choice may not always be the right one, and adding additional complexity does not always translate to better results, hence, highlighting the importance of recognizing and analysing the properties of the data to be modelled.

5.2 Error Analysis

As part of the error analysis, the words with the highest weight coefficients in the trained tf-idf LR regression model were identified. These are the words that contribute the most to the classification decision of the model. Identifying these words gives insight into which words are most important in each class. As an illustrative example, the two word-clouds in Figures 4 and 5 show the top 20 words for the Anxiety and Depression subreddit classes respectively. The word-clouds suggest that the LR tf-idf model effectively captured words that are specific for one subreddit and are thus the valuable words in differentiating this class from another.

Figure 4: Top 20 words for Anxiety.

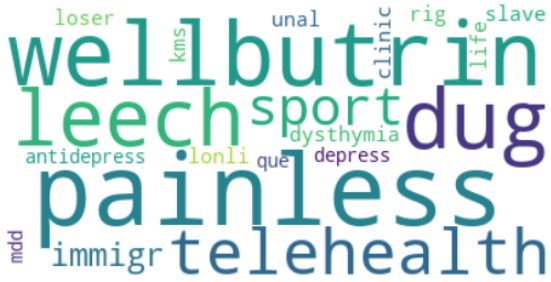

Figure 5: Top 20 words for Depression.

The predictions made by the best performing LR model were additionally evaluated by identifying typical errors made by the model, in order to gain insights into the challenges the model might face when making predictions. We observed that the model faced difficulties making accurate predictions when encountering words that are also a class label themselves or are associated with other labels within the text. To outline an example, Table 3 shows two posts from the subreddit Depression, where the first text, under "Best prediction", shows the post that had the highest predicted probability of being labelled Depression and were thus classified correctly. Conversely, the second text, under "Worst prediction", is the post that had the lowest predicted probability of being labelled Depression, and were thus mislabelled as, in this case, BPD. When examining the first text, it becomes evident that the high probability likely comes from the fact that the word "depressed" itself is mentioned in the text. Conversely, by reading the second text, it seems reasonable why the model classified the posts as BPD rather than depression, as the word "BPD" appears in the text. This example illustrates the challenge of accurately classifying certain Reddit posts, where the presence of frequent words like "BPD" can lead to a higher probability of being assigned to the "BPD" label, despite belonging to another class.

These challenges are especially an issue for the Mental Illness class, that had a very low F1 score of .40 compared to the other classes, for even the best performing model. The Mental Illness subreddit contains posts about various mental-health related topics, including Depression, BPD, Anxiety, Bipolar and Schizophrenia, which are all themselves classes in this task. The model, therefore, in many cases fails to identify patterns that are unique for the Mental Illness class, especially in cases where "Mental Illness" itself is not mentioned. Consequently, having a separate Mental Illness class that comprises all other classes adds noise to the model, and results in a high level of confusion in its predictions, as can be seen in the "Actual Mental Illness" row of the confusion matrix for the tf-idf LR model in Appendix A. Here

Table 3: Best & worst classification for depression.

Actual Label: Depression	
Best prediction	
Original text:	*"Depressed gf how do i help someone not feel depressed when i'm also depressed"*
Predicted label:	Depression
Probability of actual label:	0.99
Worst prediction	
Original text:	*"I want to buy a motorcycle I'm afraid if I get one my fp with bpd might cause me to wreck or kill myself."*
Predicted label:	BPD
Probability of actual label:	4.01e-10

we see that there are more predictions for the other classes than for Mental Illness. The classification task at hand is already difficult in general, considering the overlapping symptoms and hence words of the different mental-health classes. For the Mental Illness class, both symptoms and general classes are overlapping, which makes it even harder for the model to accurately classify a post as belonging to this class. On another note, the inclusion of this class is a good reflection of reality as there are no limits to either name or number of subreddits on the actual Reddit platform, resulting in countless unstructured forums.

5.3 Task Ambiguity & Practical Implications

Language is famously hard to model, and therefore we also must take our performance assessments with a grain of salt when creating language models. The best performing model has a weighted average F1 score of .77, which could appear low compared to many non-linguistic models that are said to be well performing. Due to the nature of language being hard, a lower apparent

performance could also be expected. We can say apparent performance because there is an element of ambiguity in our task and therefore in our performance assessment. This ambiguity arises because the task that we have set the language model up to perform, does not 100% align with the data that we have used as input. The data is labelled according to the community, that a user decides they would like to post to, and thus who the user has decided they would like to connect with and get feedback from. The task that we have assigned is to take the language used in these posts and categorize to which class the post would best belong, not to which the user will get best feedback or engagement. These two ideas may however not always line up. For example, a user with BPD could write a post heavily about their anxiety or depression symptoms, with nothing specifically talking about BPD, but post it in the BPD subreddit, as this is their community that they connect with and feel safe talking to. The feedback that the user gets on the post in the BPD subreddit is also likely to be vastly different to that which they would get if they had submitted the same post in the depression subreddit. However, this is not something that we are able to model or predict from just a short post. The same goes for number of users i.e., Redditors; not all subreddits may have large communities, thus, users may intentionally post in forums with larger crowds, rather than solely basing the decision on an appropriate subreddit name. This is very likely what has occurred in the example in Table 3, where a user has a connection to the Depression subreddit and submits there, even though their post appears to centre on BPD. Therefore, we should be critical in how we assess the performance with regards to the specific task and implementation.

This leads onto the question of practical implications, and questions of how this model could be implemented in a real-life setting. As just discussed, there is often a mismatch between what we can model, i.e., what is explicitly written in a Reddit post, and where the post has been submitted, as this is impacted by more factors than just the content of the post. These factors include implicit attributes such as which communities the user engages in frequently, which diagnoses the user has (for this specific dataset), and what type of feedback or help the user is hoping for. Hence, recommending the implementation of this model as a be-all and end-all solution that automatically categorizes posts before submission would be unwarranted. Instead, it could serve as an app add-on, offering valuable assistance in the categorization process, by bringing up suggestions of other subreddits that the user could post, to which the user may not have known about or have explored before. This could add real-life value for the user, as they may end up exploring and posting to new subreddits where they gain new and different perspectives from, and in the realm of mental health disorders, they may even have their eyes opened to new disorders to investigate and research.

A further practical implication that should be considered when dealing with internet scraped data and social media platforms, is the inherent noisiness of the data. While much of the text and information makes sense and gives extra context when we as humans read social media posts, it will often just create extra noise when structured into data. This will however always happen, when looking at sources where anyone is free to post anything they like, and therefore should be dealt with as part of the processing pipeline and should also be taken into account when assessing performance and outcomes.

5.4 Limitations

It is imperative to acknowledge the inherent limitations associated with the utilization of Natural Language Processing (NLP) and machine learning models in the given setting, as well as any other. The performance of our classification models may be prone to inaccuracies owing to several limitations, including class imbalance, limited generalizability, and the general ambiguity of language. Hence, it is crucial to carefully consider these limitations, when interpreting the results of

our study, and when extending our approach to different contexts.

Despite implementing appropriate undersampling techniques, class imbalance might still be an issue. This disparity can result in models that are biased towards the majority class, potentially resulting in lower performance for the smaller classes, as also witnessed in the given study by their class specific performance measures. As a result, the limited representation of the smaller classes, such as Bipolar and Schizophrenia, may lead to difficulties in capturing the nuances of mental health discussions related to these disorders, impacting the overall performances of the classifications. Furthermore, undersampling drastically decreases the information available for the models for learning when so many rows are discarded, potentially decreasing the performance of all classes. To address this issue, an alternative approach, such as oversampling, could be considered to balance the representation of the smaller classes.

Another constraint is the lack of coverage of mental illnesses in our dataset. While we have included subreddits for a few mental health illnesses like Depression, Anxiety, Bipolar Disorder, and Schizophrenia, our dataset does not cover the entire spectrum of mental disorders. The absence of specific diseases limits the generalizability of our method to other mental health conditions not covered in the dataset, possibly excluding individuals who may seek help or information related to such disorders.

Moreover, due to the overlapping symptoms of numerous conditions, it is important to note that categorizing mental health subreddits is generally a difficult task. Anxiety and depression, for example, are symptoms of numerous mental health illnesses, and people may discuss these symptoms across multiple subreddits, which could possibly result in ambiguous and incorrect classifications. To address this, allowing for the possibility of one post belonging to multiple subreddits or having multiple labels could provide a more nuanced representation of the complex nature of online discussions around mental health.

6 Conclusion & Future Work

The aim of this project was to investigate the impact of tf-idf and Word2Vec on the performance of LR and MLP models in classifying reddit posts related to mental health. The LR with tf-idf feature extraction outperformed all other combinations for the task, with a weighted average F1-score of .77, and a lower computational complexity. These findings show that the choice of feature extraction technique has a significant impact on the performance and computational complexity of the models. Additionally, the results indicate that the most complex model does not necessarily perform the best, highlighting the importance of identifying the best approach individually for a given use case.

In conclusion, the informal nature of user-generated text on platforms such as Reddit, and the general complexities of language, pose significant challenges for leveraging machine learning and deep learning in this field. Despite these challenges, the findings of this study give a valuable contribution in helping users to connect to the right communities regarding mental-health related concerns online.

For future work, there are several aspects that could be worth exploring further. First, extending the classification model to languages other than English would greatly contribute to its applicability and reach. Second, expanding the model to classify a broader range of mental health categories would provide a more comprehensive analysis of mental health-related text. Lastly, to address the imbalance in the dataset, further oversampling techniques could be explored. One potential technique is back-translation to augment the training data (Bonthu et al., 2022). This could be beneficial for the overall performance and ability to generalize well for unseen text data, even for classes that are small by nature.

7 References

Bonthu, S., Dayal, A., Lakshmi, M. S., & Rama Sree, S. (2022). *Effective Text Augmentation Strategy for NLP Models*. 521–531. https://doi.org/10.1007/978-981-16-4538-9_51

Cahyani, D. E., & Patasik, I. (2021). Performance comparison of tf-idf and word2vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics*, *10*(5), 2780–2788. https://doi.org/10.11591/EEI.V10I5.3157

Chen, Z., Yang, R., Fu, S., Zong, N., Liu, H., & Huang, M. (2023). Detecting Reddit Users with Depression Using a Hybrid Neural Network. *ArXiv*. https://arxiv.org/abs/2302.02759v1

Cohan, A., Soldaini, L., Desmet, B., MacAvaney, S., Yates, A., & Goharian, N. (2018). SMHD: A Large-Scale Resource for Exploring Online Language Usage for Multiple Mental Health Conditions. *COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings*, 1485–1497. https://arxiv.org/abs/1806.05258v2

Dinu, A., & Moldovan, A.-C. (2021). *Automatic Detection and Classification of Mental Illnesses from General Social Media Texts* (pp. 358–366). https://doi.org/10.26615/978-954-452-072-4_041

Géron, A. (2022). Hands-on Machine Learning with Scikit-Learn, Keras, and Hands-On Machine Learning TensorFlow. In *O Reilly Media, Inc.* (3rd ed.). O'Reilly Media, Inc. https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/

Jiang, T., Jia, L., Wan, M. C., & Meng, J. H. (2020). The Text modeling method of Tibetan text combining Word2vec and improved TF-IDF. *Journal of Physics: Conference Series*, *1601*(4). https://doi.org/10.1088/1742-6596/1601/4/042007

Jurafsky, D. & Martin, J. (2023). Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition. 3rd Edition draft.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, *18*(17), 1–5. http://jmlr.org/papers/v18/16-365.html

Ma, Y., & He, H. (2013). Imbalanced Learning: Foundations, Algorithms, and Applications. *Imbalanced Learning: Foundations, Algorithms, and Applications*, 205. https://www.wiley.com/en-sg/Imbalanced+Learning%3A+Foundations%2C+Algorithms%2C+and+Applications-p-9781118074626

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net.

Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2).

Sekulić, I., & Strube, M. (2020). Adapting Deep Learning Methods for Mental Health Prediction on Social Media. *W-NUT@EMNLP 2019 - 5th Workshop on Noisy User-Generated Text, Proceedings*, 322–327. https://doi.org/10.18653/v1/D19-5542

Vimal, B., Sem, V. I., & Anupama Kumar, S. (n.d.). *Application of Logistic Regression in Natural Language Processing*. Retrieved May 23, 2023, from www.ijert.org

WHO. (2022). World Mental Health report. *World Health Organization*, 1–260.

Yang, L., & Shami, A. (2020). On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Yates, A., Cohan, A., & Goharian, N. (2017). Depression and Self-Harm Risk Assessment in Online Forums. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2968–2978. https://doi.org/10.18653/v1/d17-1322

8 Appendix

Appendix A *– Confusion matrices on test data*

Table 1. LR - Word2Vec

		Predicted					
		BPD	**Anxiety**	**Depression**	**Mental Illness**	**Bipolar**	**Schizophrenia**
Actual	**BPD**	**32693**	1947	1068	3952	1096	269
	Anxiety	3387	**24531**	741	2177	1002	275
	Depression	1157	666	**4015**	698	327	255
	Mental Illness	4600	1660	705	**15989**	1063	177
	Bipolar	2057	1005	493	1589	**1964**	375
	Schizophrenia	293	237	289	221	297	**986**

Table 2. LR– Tf-Idf

		Predicted					
		BPD	**Anxiety**	**Depression**	**Mental Illness**	**Bipolar**	**Schizophrenia**
Actual	**BPD**	**34079**	1453	865	3167	1266	198
	Anxiety	1740	**26761**	302	1941	1210	160
	Depression	830	351	**4808**	577	324	228
	Mental Illness	2778	1469	417	**18026**	1398	110
	Bipolar	1355	885	402	1490	**2983**	372
	Schizophrenia	160	154	202	184	245	**1381**

Table 3. MLP - Word2Vec

		Predicted					
		BPD	**Anxiety**	**Depression**	**Mental Illness**	**Bipolar**	**Schizophrenia**
Actual	**BPD**	**31076**	1738	2053	3848	1535	775
	Anxiety	2689	**23581**	1239	2139	1646	819
	Depression	716	438	**4633**	581	252	498
	Mental Illness	3901	1521	1373	**15668**	1279	452
	Bipolar	1459	793	835	1502	**2182**	712
	Schizophrenia	168	131	375	162	178	**1309**

Table 4. MLP – Tf-Idf

		Predicted					
		BPD	**Anxiety**	**Depression**	**Mental Illness**	**Bipolar**	**Schizophrenia**
Actual	**BPD**	**29977**	2523	2021	4908	1119	480
	Anxiety	3475	**22570**	1244	3428	903	494
	Depression	1066	587	**4127**	812	221	305
	Mental Illness	4288	2104	1127	**15511**	882	286
	Bipolar	1994	1198	607	1835	**1390**	463
	Schizophrenia	255	252	372	254	150	**1043**