Kathmandu University Journal of Science, Engineering and Technology, Vol. 19, No. 1, March 2025

Kathmandu University
Journal of Science, Engineering and Technology

Enhancing disease prediction from limited datasets

Ayush Singh *

Department of Mathematics, Kathmandu University, Dhulikhel, Kavre, Nepal.

Abstract

In this paper, the enhancement of disease prediction accuracy for a limited dataset is explored. Text representation models such as ClinicalBERT and TF-IDF vectorizer are utilized to generate text embeddings, which are then paired with robust classification algorithms (estimators), including Random Forest, XGBoost, and linear models like the Passive Aggressive Classifier. While embeddings of advanced text representation models combined with robust classification algorithms are expected to yield satisfactory results, this research focuses on comparing two different text representation models and how the text embeddings they generate perform when combined with estimators in predicting diseases. Additionally, the compatibility of text representation models with classification algorithms, and its impact on accuracy for disease prediction in the limited dataset is examined.

Keywords: Text representation models; Classification algorithms; Compatibility; Text embeddings.

1. Introduction

The application of Natural Language Processing (NLP) [1] in disease prediction began in the early 2000s, when computational methods were first explored by the medical community to extract valuable insights from unstructured medical data, such as clinical notes and patient narratives [2]. Over time, advancements in computational power and NLP techniques have facilitated the identification of early disease markers and improved diagnostic accuracy through the analysis of vast amounts of previously untapped clinical text data [3].

A significant challenge in the medical field has been the underutilization of unstructured data, which often remains inaccessible in free-text formats like doctors' notes and patient narratives. In the absence of NLP techniques, much of this critical information goes unexamined, leading to less accurate diagnoses, treatment delays, and inefficiencies in patient care [4].

Solutions such as ClinicalBERT or GPT-based models [4] are now being employed to process unstructured data, identify symptoms, and conduct predictive analyses based on patient narratives [5]. BERT [6] can be utilized as a feature extractor, where patient symptom descriptions are converted into high-quality numerical representations (embeddings) rather than being directly classified. The embeddings generated by BERT capture the context, meaning, and relationships between words in the text. By providing dense, context-rich embeddings, BERT significantly enhances the quality of input features for classification [7]. Conversely, TF-IDF generates a sparse matrix representing the importance of words within documents [8]. This hybrid approach [9] emphasizes the strengths of text representation models in understanding natural language and classification models in making accurate predictions from structured data (text embeddings).

The work done on [7] is focused on fine-tuning ClinicalBERT and assessing the quality of biomedical embeddings by analyzing correlations between doctor-rated relationships and embedding similarity scores. In this research [9], an efficient multiclass model using XGBoost [10] has been established to help identify tumor origins in a data group comprising 10 types of tumor based on copy number variations. The XGBoost classifier is then applied to these 10 types of cancer, achieving superior accuracy regardless of the training datasets or the independent validation dataset by selecting 300 features, outpacing four other classifiers: KNN [11], DNN [12], SVM [13], and Adaboost [14]. Furthermore, [15] examines tree-based models [16] like XGBoost and RandomForest [17] for multiclass classification involving Alzheimer's disease (AD), normal cognition (NC), and mild cognitive impairment (MCI). In addition, embedded files from electronic health records are used to improve prediction tasks. Embedding algorithms are increasingly utilized to depict clinical concepts in healthcare, thereby improving machine learning tasks such as clinical phenotyping and disease prediction. [18] This study introduces a pre-training scheme for longitudinal healthcare data, creating universal medical concept embeddings. The research systematically evaluates various models, including tree-based methods across ten patient-level prediction tasks. Findings indicate that integrating pre-trained embeddings with tree-based models enhances prediction accuracy in tasks such as adverse event detection and disease risk assessment. [19] Med-BERT adapts the BERT framework to structured EHR data, generating contextualized embeddings from a dataset of over 28 million patients. The study demonstrates that these embeddings, when used with models like LightGBM [20], significantly improve disease prediction accuracy, especially in scenarios with limited training data . Although no specific literature was found on the application of ClinicalBERT embeddings combined with estimators to predict various number of diseases, there is substantial evidence that the use of embeddings with tree-based models may enhance the precision of disease prediction [21].

This work focuses on comparing the performance of two text representation models (ClinicalBERT and TF-IDF) in combination with tree based models (Random Forest, XGBoost). They are evaluated on different metrics [22] (Accuracy, Precision, recall, f1-score)

*Corresponding author. Email: timgma09876@gmail.com for predicting diseases in our dataset. Additionally, a linear model [23] (Passive Aggressive Classifier [24]) is employed to contrast the performance of tree-based models with that of linear models. This work mainly focuses on combining the advanced text representation model ClinicalBERT with robust classification algorithms to see how it can enhance disease prediction for the limited dataset. To show this comparison, TF-IDF, a traditional statistical method is used, to evaluate how the rich contextual embeddings produced by ClinicalBERT might outperform the traditional approach [25].

2. Materials and methods

2.1. Dataset

Free-text patient narratives containing symptoms. The dataset [26] contains 4234 records and has over 141 disease categories. The features of dataset are as follows.

- Unnamed: 0 – an index column that does not have analytical significance.
- drugName – Name of the drug being reviewed.
- condition – Medical condition for which the drug was prescribed.
- review – Textual review by a user about their condition and experience with the drug.
- rating – Numerical rating (likely on a scale of 1-10) representing user satisfaction.
- date – Date when the review was posted.
- usefulCount – Number of times the review was marked as useful by other users.

Among these features condition and review is focused for this work as shown in Table 1. All the other features are dropped. Additionally, in this dataset, Birth Control is included as a disease category, so this is also removed. Hence, 140 disease categories are classified. The distribution of data is given in Fig. 1.

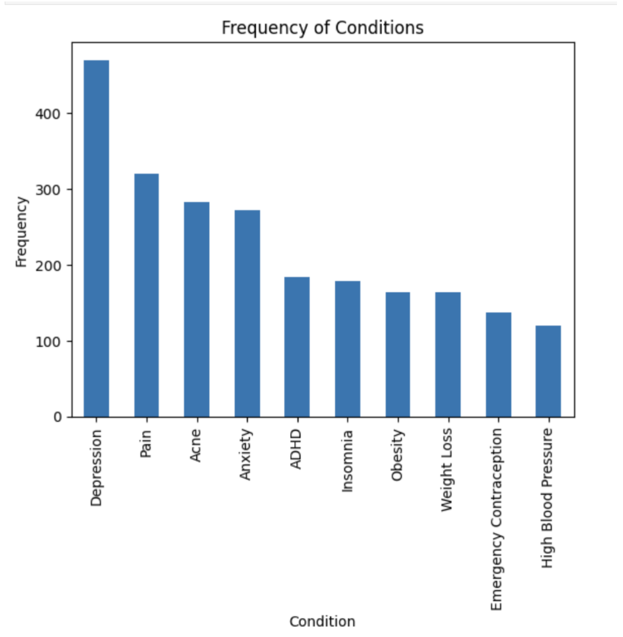


Figure 1: Distribution of data.

2.2. Data preprocessing

2.2.1. Cleaning the data

Cleaning text is the first and crucial step in text preprocessing, as raw text data can be noisy and contain irrelevant content. One of the first things to address is removing HTML tags, , special characters and symbols like (@, $, %), or even punctuation marks often don't contribute to the analysis and should be removed. Regular expressions (regex) [27] can be utilized here to identify and eliminate these characters.

Table 1: Sample conditions and reviews.

Condition	Review
Left ventricular dysfunction	"It has no side effect, I take it in combination with Bystolic 5 Mg and Fish Oil."
ADHD	"My son is halfway through his fourth ek of Intuniv. After experiencing significant mood swings with stimulants, this has been an ansr to our prayers."
Birth control	"I used to take another oral contraceptive, which gave me constant mood swings and ight gain. This one is much better."
Birth control	"This is my first time using any form of birth control. So far, I haven't had any issues except for minor headaches."
Opiate dependence	"Suboxone has completely turned my life around. I no longer crave opioids, and I feel like myself again."

2.2.2. Normalization

Normalization standardizes the text, ensuring that variations in words or characters do not cause discrepancies during analysis. The first step in this process is converting all the text to lowercase. A key step in Normalization is Lemmatization [28]. Lemmatization uses a sophisticated approach, understanding the context and grammar of the word to return a valid base form, like converting "better" to "good." Tools like NLTK and SpaCy [29] provide built-in lemmatization functions that apply these transformations as shown in Fig. 2

2.2.3. Tokenization

Tokenization is the process of splitting text into smaller, manageable pieces called tokens. These tokens can be words, sentences, or subword units. The most common form is word tokenization, where the text is split into individual words. This helps models process text by focusing on one token at a time. Tokenization can be done using tools like SpaCy's tokenizer [30].

2.2.4. Vectorization

After the text has been cleaned and tokenized, the next step is to convert it into a numerical format that machine learning algorithms can process. This is called vectorization [31], and it involves transforming text into numerical representations such as word counts, term frequencies, or dense word embeddings. The vectorization used is Term Frequency- Inverse Document Frequency (TF-IDF).

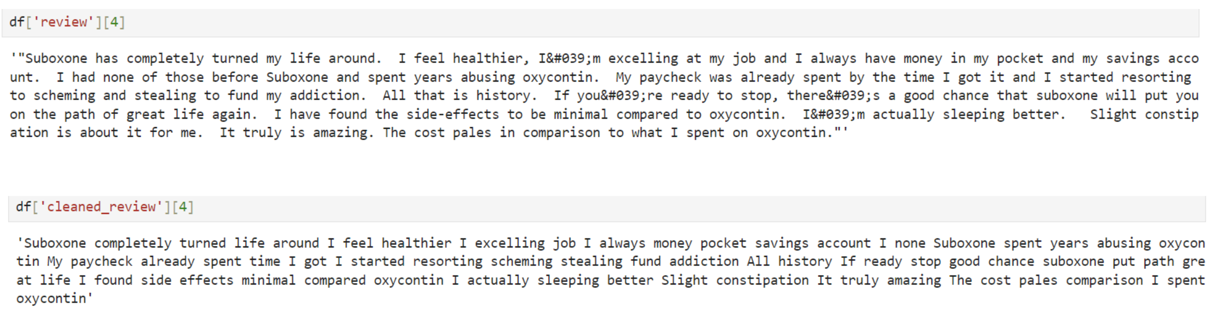
2.3. Implementation of text representation models

2.3.1. Term frequency-inverse document frequency (TF-IDF)

TF-IDF is a way to figure out how important a word is in a document compared to a bunch of other documents.

$$\text{TF-IDF}(t,d,D) = \text{TF}(t,d) \times \text{IDF}(t,D) \tag{1}$$

- D: Collection of documents



```
df['review'][4]
```

```
'"Suboxone has completely turned my life around.  I feel healthier, I&#039;m excelling at my job and I always have money in my pocket and my savings acco
unt.  I had none of those before Suboxone and spent years abusing oxycontin.  My paycheck was already spent by the time I got it and I started resorting
to scheming and stealing to fund my addiction.  All that is history.  If you&#039;re ready to stop, there&#039;s a good chance that suboxone will put you
on the path of great life again.  I have found the side-effects to be minimal compared to oxycontin.  I&#039;m actually sleeping better.   Slight constip
ation is about it for me.  It truly is amazing. The cost pales in comparison to what I spent on oxycontin."'
```

```
df['cleaned_review'][4]
```

```
'Suboxone completely turned life around I feel healthier I excelling job I always money pocket savings account I none Suboxone spent years abusing oxycon
tin My paycheck already spent time I got I started resorting scheming stealing fund addiction All history If ready stop good chance suboxone put path gre
at life I found side effects minimal compared oxycontin I actually sleeping better Slight constipation It truly amazing The cost pales comparison I spent
oxycontin'
```

Figure 2: Normalized text.

- d: A certain document
- t: Term in document d

Term Frequency It checks how often a word shows up in one document. Words that appear more often are considered more important.

$$\text{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d} \tag{2}$$

Inverse Document Frequency It looks at all the documents and checks how common a word is. If a word appears in many documents, it's considered less special or important.

$$\text{IDF}(t,D) = \log\left(\frac{\text{Total number of documents } D}{\text{Number of documents containing term } t}\right) \tag{3}$$

First the dataset is tokenized into words. The frequency of each word in document is calculated (Term Frequency), the Inverse Document Frequency is calculated based on how frequently the word appears across all the documents. The TF-IDF score is computed as the product of TF and IDF for each word. The output is a sparse matrix where each document is represented as a vector, with the weight of each word indicating its importance to that specific document [8].

2.3.2. ClinicalBERT

ClinicalBERT is a specialized version of the BERT model, fine-tuned specifically for medical and clinical text. While standard BERT is trained on general language data, ClinicalBERT focuses on capturing the unique language used in healthcare settings, such as electronic health records (EHRs), patient notes, and medical literature. ClinicalBERT recognizes "fever" and "high temperature" hold the same meaning and place them close to each other [32].

The text data is tokenized and passed through the ClinicalBERT model, it transforms the text into dense, contextualized embeddings that capture the semantic meaning of the medical text [7].

Mathematical representation of ClinicalBERT embeddings Let T represent the sequence of tokens in a clinical text input, such that:

$$T = \{t_1, t_2, \ldots, t_n\}$$

where n is the number of tokens.

Token embeddings Each token t_i is converted into an embedding $e_i \in \mathbb{R}^d$, where d is the embedding dimension (e.g., 768 for the base ClinicalBERT model):

$$e_i \in \mathbb{R}^d$$

The embeddings are made by using 3 components :

e_i = Word embeddings + Position embeddings + Segment embeddings.

Transformer layers ClinicalBERT applies a series of transformer layers $L_1, L_2, \ldots, L_k$, where k is the number of layers (e.g., 12 for BERT-base). These layers are responsible for undestanding the relationship between tokens and building a contextual representation of text . The hidden state at the l-th layer for token t_i is represented as $h_i^{(l)} \in \mathbb{R}^d$:

$$h_i^{(l)} \in \mathbb{R}^d$$

Contextual embeddings The output embeddings after the final layer L_k for each token are the dense, contextually aware embeddings:

$$h_i^{(k)} = f_k(e_i, e_1, e_2, \ldots, e_n)$$

where f_k represents the transformer operations (multi-head attention and feed-forward layers).

Each layer adds more context to the token embeddings, allowing ClinicalBERT to capture subtle meanings in clinical context.

Pooled sentence embedding After passing through the final transformer layer, $[CLS]$ embeddings contain summary of entire input text, based on the relationship and context learned by the model. Often, the embedding for the special classification token $[CLS]$, denoted as $h_{\text{[CLS]}}^{(k)}$, is used as a dense representation of the entire input text:

$$E = h_{\text{[CLS]}}^{(k)} \in \mathbb{R}^d$$

This vector E serves as the final dense embedding for the input text, summarizing its semantic and contextual meaning [33].

2.4. Implementing estimators

2.4.1. XGBoost (Extreme gradient boosting)

XGBoost is an ensemble learning method based on gradient boosting, designed for high performance and efficiency. It builds multiple decision trees sequentially, where each new tree corrects the errors of the previous ones. The model minimizes a loss function using gradient descent while applying regularization techniques to prevent overfitting.

The prediction in XGBoost is computed as:

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

where $f_k(x)$ represents the output of the k^{th} tree.

The objective function consists of two parts: a loss function L that measures the model's error and a regularization term Ω to control complexity:

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

XGBoost is widely used for structured data and provides high accuracy while being computationally efficient [10].

2.4.2. Random forest classifier

Random Forest is an ensemble learning technique that constructs multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. It follows the bagging (Bootstrap Aggregation) method, where random subsets of data and features are used for training each tree independently.

For classification, the final prediction is determined by majority voting:

$$\hat{y} = \arg\max \sum_{i=1}^{N} h_i(x)$$

where $h_i(x)$ represents the prediction from the i^{th} tree.

Random Forest is effective for handling non-linear relationships and is robust against overfitting due to its multiple decision trees [17].

2.4.3. Passive-aggressive classifier

The Passive-Aggressive Classifier is an online learning algorithm, meaning it updates itself dynamically as new data arrives. Unlike batch learning models, which train on the entire dataset at once, Passive-Aggressive updates only when it makes a wrong prediction. Working:

- If the model classifies an instance correctly, it remains passive (no update).
- If the model misclassifies, it updates its weights aggressively to correct the mistake.

The loss function used in Passive-Aggressive learning is:

$$L(w) = \max(0, 1 - y_i w^T x_i)$$

where:

- w is the weight vector,
- y_i is the true class label (+1 or -1),
- x_i is the feature vector.

When an error occurs, the model updates its weight vector as follows:

$$w = w + \eta y_i x_i$$

where η is the learning rate.

Passive-Aggressive Classifiers are particularly useful for real-time learning tasks such as spam detection, sentiment analysis, and fraud detection [34].

2.5. Evaluation metric

The classification_report function from the Scikit-learn library is a powerful tool for evaluating classification models. It provides a comprehensive summary of various performance metrics for each class in a classification problem, including precision, recall, F1-score, and support. Below is a detailed explanation of these metrics and how to interpret the output of the classification report [22].

- Precision: Indicates the proportion of positive identifications (model predicted class 1) which were actually correct. A model which produces no false positives has a precision of 1.0:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

- Recall: Indicates the proportion of actual positives which were correctly classified. A model which produces no false negatives has a recall of 1.0:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

- F1-score: Harmonic mean of Precision and Recall:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Accuracy: The overall accuracy of the model across all classes.
- Support: The number of samples each metric was calculated on.
- Macro avg: The average performance across classes without considering class imbalance.
- Weighted average: The average performance across classes that accounts for class imbalance by weighting each class's score by its support.

3. Results and discussion

The initial hypothesis suggested that combining advanced text representation models with robust classification algorithms would yield satisfactory results. However, this was not the case. In fact, the TF-IDF method as shown in Table 5, 6, 7 which is known for its simplicity and efficiency but lacks contextual awareness, outperformed ClinicalBERT as shown in Table 2, 3, 4. Although ClinicalBERT is designed to understand medical terminology and relationships, the dense feature matrix it encodes is not suitable to use with tree-based estimators like RandomForest and XGBoost.

Despite the rich contextual embeddings produced by ClinicalBERT, they are not ideal inputs for these estimators. Additionally, when working with smaller datasets, complex models such as ClinicalBERT tend to underperform since they require large amounts of data to fully utilize their context-aware capabilities. Furthermore, the distribution of data in the dataset is uneven, leading to some disease categories having a limited number of patient narratives. BERT requires extensive fine-tuning and tree based models require hyperparameter tuning to work optimally which can be computationally expensive and time consuming. While BERT embeedings capture semantic meaning but might not generalize well for domain-specfic texts like patient reviews.

Surprisingly, TF-IDF outperformed ClinicalBERT because the sparse matrix it produces is well-suited for linear classifiers, such as the Passive Aggressive Classifier. Passive Aggressive Classifiers are simple and efficient linear classifiers that work effectively with sparse data like TF-IDF matrices. For smaller datasets, TF-IDF appears to generalize text better than ClinicalBERT. Passive Aggressive Classifier is desgined for online learning, which makes it robust to class imbalance.

4. Conclusion

The result highlights that the choice of text representation and classification algorithm must be properly aligned for optimal performance. While ClinicalBERT provides rich semantic information, its effectiveness depends on how well the classifier can leverage that information. This explains why simpler models like TF-IDF, when paired with a compatible classifier such as the Passive Aggressive Classifier, outperformed ClinicalBERT + XGBoost in our experiments. The accuracy of TF-IDF + Passive Aggressive Classifier was 62.57%, while the accuracy of ClinicalBERT + XGBoost was just 25.5%. The low accuracy observed can be attributed to several factors. Certain disease categories have significantly more samples than others, which may lead to a model biased towards the majority class, thus resulting in poor performance on minority classes. For instance, in the case of ClinicalBERT + XGBoost, high blood pressure had only 4 samples. Furthermore, the patient narratives often contain subjective language, potentially causing the model to struggle in associating symptoms with diseases. Additionally, the dataset is relatively small, with only 4,234 records across 140

Table 2: Performance metrics for ClinicalBERT + XGBoost.

Disease / Metric	Accuracy	Precision	Recall	F1-score	Support
ADHD		0.15	0.15	0.15	41
High blood pressure	25.5%	0.19	0.18	0.19	4
Migraine		0.27	0.19	0.22	21
Macro avg		0.09	0.07	0.08	–
Weighted avg		0.22	0.25	0.22	–

Table 3: Performance metrics for ClinicalBERT + passive aggressive classifier.

Disease / Metric	Accuracy	Precision	Recall	F1-Score	Support
ADHD		0.44	0.29	0.35	41
High blood pressure	34.83%	0.50	0.50	0.50	4
Migraine		0.50	0.52	0.51	21
Macro avg		0.13	0.11	0.11	–
Weighted avg		0.37	0.35	0.33	–

Table 4: Performance metrics for ClinicalBERT + random forest.

Disease / Metric	Accuracy	Precision	Recall	F1-Score	Support
ADHD		0.30	0.22	0.25	41
High blood pressure	30.22%	0.20	0.04	0.06	4
Migraine		0.40	0.15	0.22	21
Macro avg		0.11	0.08	0.07	–
Weighted avg		0.26	0.30	0.22	–

Table 5: Performance metrics for TF-IDF + XGBoost.

Disease / Metric	Accuracy	Precision	Recall	F1-Score	Support
ADHD		0.87	0.77	0.82	35
High blood pressure	57.68%	0.62	0.67	0.64	27
Migraine		0.86	0.78	0.82	23
Macro avg		0.30	0.25	0.26	–
Weighted avg		0.56	0.58	0.55	–

Table 6: Performance metrics for TF-IDF + passive aggressive classifier.

Disease / Metric	Accuracy	Precision	Recall	F1-Score	Support
ADHD		0.83	0.86	0.85	35
High blood pressure	62.57%	0.65	0.63	0.64	27
Migraine		0.83	0.83	0.83	23
Macro avg		0.44	0.41	0.41	–
Weighted avg		0.60	0.63	0.60	–

Table 7: Performance metrics for TF-IDF + random forest.

Disease / Metric	Accuracy	Precision	Recall	F1-Score	Support
ADHD		0.73	0.77	0.75	35
High blood pressure	58.68%	0.67	0.59	0.63	27
Migraine		0.81	0.91	0.86	23
Macro avg		0.32	0.25	0.26	–
Weighted avg		0.56	0.59	0.53	–

categories, which hinders the classification model's ability to generalize effectively. To enhance performance, ClinicalBERT can be fine-tuned on the dataset to better adapt it to the domain-specific text. Balancing the dataset through oversampling or undersampling techniques could also help reduce the bias towards majority classes. Some limitations of our current model include the limited dataset size, impacting model generalization. Moreover, the deep learning approach of ClinicalBERT requires high processing power, consumes significant memory, and requires longer training times, making simpler, more efficient models a better trade-off. Thus, if the dataset is limited and imbalanced, TF-IDF + Passive Aggressive Classifier serves as a strong baseline model.

References

[1] IBM. What is natural language processing? (2024). URL https://www.ibm.com/think/topics/natural-language-processing.

[2] Jaotombo F, Adorni L, Ghattas B & Boyer L, Finding the best trade-off between performance and interpretability in predicting hospital length of stay using structured and unstructured data, *PLOS ONE*, 18(11) (2023) e0289795. ISSN 1932-6203. https://doi.org/10.1371/journal.pone.0289795.

[3] Omar M, Brin D, Glicksberg B & Klang E, Utilizing natural language processing and large language models in the diagnosis and prediction of infectious diseases: A systematic review, *American Journal of Infection Control*, 52(9) (2024) 992–1001. ISSN 0196-6553. https://doi.org/10.1016/j.ajic.2024.03.016.

[4] Huang K, Altosaar J & Ranganath R, ClinicalBERT: Modeling clinical notes and predicting hospital readmission, *arXiv preprint arXiv:1904.05342*. https://doi.org/10.48550/arXiv.1904.05342.

[5] Lu M, Jin X & Wang Z. ClinicalBertSum: RCT summarization by using clinical BERT embeddings (2020). URL https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/reports/custom/report29.pdf, stanford CS224N Final Project.

[6] IBM. How BERT and GPT models change the game for NLP (2024). URL https://www.ibm.com/think/insights/how-bert-and-gpt-models-change-the-game-for-nlp.

[7] Alsentzer E, Murphy J R, Boag W, Weng W H, Jin D, Naumann T & McDermott M B A. Publicly available clinical BERT embeddings (2019). https://doi.org/10.48550/ARXIV.1904.03323.

[8] GeeksforGeeks. Understanding TF-IDF (Term frequency-inverse document frequency) (2021). URL https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/, educational Blog.

[9] Zhang Y, Feng T, Wang S, Dong R, Yang J, Su J & Wang B, A novel XGBoost method to identify cancer tissue-of-origin based on copy number variations, *Frontiers in Genetics*, 11. ISSN 1664-8021. https://doi.org/10.3389/fgene.2020.585029.

[10] *XGBoost: A Scalable Tree Boosting System*. arXiv (2016). https://doi.org/10.48550/ARXIV.1603.02754.

[11] W3Schools. *Python Machine Learning - KNN* (2024). URL https://www.w3schools.com/python/python_ml_knn.asp.

[12] TutorialsPoint. Python deep learning - Deep neural networks (2024). URL https://www.tutorialspoint.com/python_deep_learning/python_deep_learning_deep_neural_networks.htm.

[13] Scikit-learn. Support vector machines (2024). URL https://scikit-learn.org/stable/modules/svm.html.

[14] DigitalOcean. A guide to AdaBoost: Boosting to save the day (2024). URL https://www.digitalocean.com/community/tutorials/adaboost-optimizer.

[15] Lin W, Gao Q, Du M, Chen W & Tong T, Multiclass diagnosis of stages of alzheimer's disease using linear discriminant analysis scoring for multimodal data, *Computers in Biology and Medicine*, 134 (2021) 104478. ISSN 0010-4825. https://doi.org/10.1016/j.compbiomed.2021.104478.

[16] Ravindran D. Tree-based machine learning algorithms explained (2024). URL https://medium.com/analytics-vidhya/tree-based-machine-learning-algorithms-explained-b50937d3cf8e.

[17] Breiman L, Random forests, *Machine Learning*, 45(1) (2001) 5–32. ISSN 0885-6125. https://doi.org/10.1023/A:1010933404324.

[18] Li Y, Dong W, Ru B, Black A, Zhang X & Guan Y, Generic medical concept embedding and time decay for diverse patient-level prediction tasks, *iScience*, 25(9) (2022) 104880. ISSN 2589-0042. https://doi.org/10.1016/j.isci.2022.104880.

[19] Rasmy L, Xiang Y, Xie Z, Tao C & Zhi D, Med-BERT: Pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction, *arXiv*. https://doi.org/10.48550/ARXIV.2005.12833.

[20] M A. Complete guide on how to Use LightGBM in Python (2021). URL https://www.analyticsvidhya.com/blog/2021/08/complete-guide-on-how-to-use-lightgbm-in-python/.

[21] Yi F, Yang H, Chen D, Qin Y, Han H, Cui J, Bai W, Ma Y, Zhang R & Yu H, XGBoost-shap-based interpretable diagnostic framework for alzheimer's disease, *BMC Medical Informatics and Decision Making*, 23(1) (2023) 22. ISSN 1472-6947. https://doi.org/10.1186/s12911-023-02238-9.

[22] Data Science Wizards. Evaluation metrics for machine learning or data models (2020). URL https://medium.com/@datasciencewizards/evaluation-metrics-for-machine-learning-or-data-models-52c277e0ff3b, medium Article.

[23] Scikit-learn. *sklearn.linear_model.PassiveAggressiveClassifier* (2024). URL https://scikit-learn.org/0.15/modules/generated/sklearn.linear_model.PassiveAggressiveClassifier.html.

[24] GeeksforGeeks, Passive aggressive classifiers, *GeeksforGeeks*. URL https://www.geeksforgeeks.org/passive-aggressive-classifiers/.

[25] Patel D, Timsina P, Gorenstein L, Glicksberg B S, Raut G, Cheetirala S N, Santana F, Tamegue J, Kia A, Zimlichman E, Levin M A, Freeman R & Klang E, Traditional machine learning, deep learning, and BERT (large language model) approaches for predicting hospitalizations from nurse triage notes: Comparative evaluation of resource management, *JMIR AI*, 3 (2024) e52190. ISSN 2817-1705. https://doi.org/10.2196/52190.

[26] invincibleaayu. AI disease prediction dataset (2022). URL https://github.com/invincibleaayu/AI-disease-prediction/tree/main/usingNLP/data, accessed: November 25, 2024.

[27] W3Schools. Python RegEx (2024). URL https://www.w3schools.com/python/python_regex.asp.

[28] TechTarget. Lemmatization in NLP (2024). URL https://www.techtarget.com/searchenterpriseai/definition/lemmatization.

[29] Jay. NLTK vs SpaCy: A deeper dive into NLP libraries (2024). URL https://medium.com/@seaflux/nltk-vs-spacy-a-deeper-dive-into-nlp-libraries-1416dbc5ac7f.

[30] GeeksforGeeks. Tokenization using Spacy library (2024). URL https://www.geeksforgeeks.org/tokenization-using-spacy-library.

[31] Dremio. Vectorization in NLP (2024). URL https://www.dremio.com/wiki/vectorization-in-nlp/#:~:text=Vectorization%20in%20NLP%20is%20used,TF%2DIDF%2C%20and%20Word2Vec.

[32] Eleventh Hour Enthusiast. Adapting BERT for biomedical and clinical NLP: ClinicalBERT and BlueBERT (2020). URL https://medium.com/@EleventhHourEnthusiast/adapting-bert-for-biomedical-and-clinical-nlp-clinicalbert-and-bluebert-64cbdc33a00b, Medium Article.

[33] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L & Polosukhin I, Attention is all you need, *arXiv preprint arXiv:1706.03762*. https://doi.org/10.48550/ARXIV.1706.03762.

[34] Crammer K, Dekel O, Keshet J, Shalev-Shwartz S & Singer Y, Online passive-aggressive algorithms, *J. Mach. Learn. Res.*, 7 (2006) 551–585. ISSN 1532-4435. URL https://jmlr.csail.mit.edu/papers/volume7/crammer06a/crammer06a.pdf.