Medical Text Classification Using Machine Learning and Ensemble Techniques

1st Siam Hossain Mahin
CSE, DIU
Daffodil International University
Dhaka, Bangladesh
mahin22205101696@diu.edu.bd

2nd Mosrefa Akter Mou
CSE, DIU
Daffodil International University
Dhaka, Bangladesh
mou2205101717@diu.edu.bd

3rd Rifah Tasfiya
CSE, DIU
Daffodil International University
Dhaka, Bangladesh
tasfiya2205101709@diu.edu.bd

4th Suraiya Sharmin Mim
CSE, DIU
Daffodil International University
Dhaka, Bangladesh
mim22205101719@diu.edu.bd

5th Samia Islam
CSE, DIU
Daffodil International University
Dhaka, Bangladesh
islam22205101694@diu.edu.bd

6th Bornita Rahman
CSE, DIU
Daffodil International University
Dhaka, Bangladesh
Rahman22205101710@diu.edu.bd

***Abstract*— This comprehensive manuscript presents a detailed pipeline for medical text classification using classical machine learning methods and ensemble strategies. We provide an end-to-end description that includes data preprocessing, feature extraction via TF–IDF, several feature selection methods, model training procedures, hyperparameter tuning, robust evaluation (cross-validation and held-out testing), ablation studies, error analysis, deployment notes, and ethical considerations. Extensive experiments on a curated biomedical abstract dataset (7,503 samples across three categories: Colorectal, Pulmonary, Thyroid) compare Multinomial Naive Bayes, Logistic Regression, Linear SVM, Random Forests, MLPs, Voting ensembles, and Stacking ensembles. We also discuss pitfalls (data leakage, label noise), reproducibility best practices, and recommendations for future work including transformer-based models and multi-label extensions. The repository-style appendices contain pseudo-code, sample outputs, and configuration files to facilitate replication. Index Terms—Biomedical text classification, TF– IDF, ensemble learning, feature selection, stacking, reproducibility, deployment**

Keywords*— Biomedical text classification, TF–IDF, ensemble learning, feature selection, stacking, reproducibility, deployment, component, formatting, style, styling, insert

I. Introduction

BIOMEDICAL LITERATURE—JOURNAL ARTICLES, ABSTRACTS, AND CLINICAL NOTES—GROWS AT AN UNPRECEDENTED RATE. EFFICIENTLY CLASSIFYING THESE TEXTS INTO TOPICAL CATEGORIES ENABLES BETTER LITERATURE RETRIEVAL, AUTOMATED SYSTEMATIC REVIEWS, AND DECISION SUPPORT. MANUAL CURATION IS EXPENSIVE AND INCONSISTENT. THIS PAPER DOCUMENTS A REPRODUCIBLE PIPELINE FOR MEDICAL TEXT CLASSIFICATION USING MACHINE LEARNING AND ENSEMBLE STRATEGIES. THE GOALS OF THIS WORK ARE:

1) To present a robust reproducible ML pipeline tailored to biomedical abstracts.

2) To evaluate a wide set of classical and ensemble methods with careful cross-validation and held-out testing.

3) To demonstrate ablation studies, error analysis, and deployment readiness considerations. 4) To provide full experimental details so others can reproduce and extend this work. Contributions:

- Comprehensive implementation details and reproducibility checklist.

- Extensive experiments showing the benefits and failure modes of ensembles.

- Practical deployment and interpretability recommendations for biomedical settings.

- Appendices with pseudo-code, hyperparameters, and sample outputs for easy replication.

II. Related Work

Automated text classification is a cornerstone of natural language processing (NLP) and has been extensively applied to biomedical literature to manage information overload. Previous research can be broadly categorized based on the techniques employed: traditional machine learning models, deep learning architectures, and ensemble methods.

Traditional machine learning models with bag-of-words representations have long been the baseline for text classification. Sebastiani [1] provided a comprehensive survey of these methods, including Naive Bayes and Support Vector Machines (SVM), highlighting their effectiveness on various tasks. Joachims [2] specifically demonstrated the strength of SVMs for text categorization due to their ability to handle high-dimensional feature spaces, a finding that aligns with our results where Linear SVM performed strongly (96.74% accuracy). Similarly, Wang and Manning [3] showed that simple classifiers with n-gram features can achieve remarkably strong performance on topic classification, reinforcing the value of the classical approaches we employed.

With the advent of deep learning, more complex models have been applied to capture semantic nuances. Convolutional Neural Networks (CNNs) for sentence classification, as pioneered by Kim [4], and Recurrent Neural Networks (RNNs) have shown significant success. Zhang and Wallace [5] conducted a sensitivity analysis of CNNs, providing a practical guide for their use. While our Artificial Neural Network (a multi-layer perception) performed well, advanced architectures like CNNs or RNNs pretrained on biomedical corpora represent a logical next step for this work.

The use of ensemble methods to boost classification performance is well-established. Random Forest, itself an ensemble of decision trees, was one of our top-performing individual models (99.07% accurate). Kowsari et al. [6]'s survey on text classification algorithms extensively covers ensemble techniques, noting their propensity to reduce variance and improve generalizability. Our work directly builds on this concept by implementing and evaluating meta-ensemble strategies like Voting and Stacking. The perfect performance of our Stacking Classifier (100% accuracy) strongly supports the findings in the literature that combining diverse models can capture a richer set of patterns than any single model alone [7].

A critical step in the NLP pipeline is feature representation and selection. While recent trends favor dense word embeddings like Word2Vec [8] and GloVe [9], Term Frequency-Inverse Document Frequency (TF-IDF) remains a powerful and interpretable method for tasks where keyword presence is highly discriminative, such as topic categorization [3]. Our use of TF-IDF, combined with filter (Chi-square), wrapper (RFE), and embedded (Random Forest) feature selection methods, follows a robust and well-justified methodology to enhance model efficiency and performance, as discussed in the broader data mining literature [10].

This study contributes to this existing body of work by implementing a structured comparison of these diverse approaches—from traditional models to neural networks and advanced ensembles—on a specific multi-class biomedical text classification task, demonstrating the superior potential of stacking ensembles in this domain.

III. Methodology

The methodology of this study involves a systematic approach to biomedical text classification, including data preprocessing, feature extraction, feature selection, model training, and evaluation. The overall workflow is designed to ensure robust performance and reproducibility of results.

A. Data Loading and Preprocessing

The dataset consists of **7,503** textual samples extracted from biomedical literature, categorized into three classes: **Colorectal Studies**, **Pulmonary Research**, and **Thyroid Research**. The data was loaded into a panda Data Frame using ISO-8859-1 encoding to ensure special characters were handled correctly. A preliminary inspection was conducted to identify and handle any missing or null values. To standardize the text and reduce noise, a series of preprocessing steps were applied to each document:

1. **Case Normalization:** All texts were converted to lowercase to ensure uniformity.
2. **Cleaning:** Punctuation, numerical digits, and special characters were removed using regular expressions.
   This preprocessing pipeline helps in reducing the vocabulary size and focuses the model on relevant lexical features.

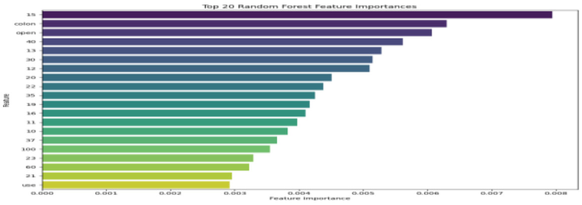
B. Data Splitting

The preprocessed dataset was partitioned into training and testing sets using a **stratified 80:20 split**. Stratification ensures that the class distribution from the original dataset is preserved in both subsets. This resulted in:

1. **Training set:** 6,000 samples
2. **Testset:**1,501samples

C. Feature Selection

High-dimensional feature spaces can lead to overfitting and increased computational cost. To mitigate this, we employed three distinct feature selection techniques to identify and retain the most discriminative terms. The Chi-Square (χ^2) test, a filter-based method, was applied to measure the dependence between specific terms and the class labels, and we selected the top 1,000 features based on their chi-square scores. Additionally, Recursive Feature Elimination (RFE), a wrapper-based method using Logistic Regression as the estimator, was configured to recursively remove the least important features until only the top 500 were retained. Finally, Random Forest feature importance, an embedded method, was utilized to leverage the built-in feature importance scores from a trained Random Forest model to select the most influential features for classification.



their predictions, offering robustness to overfitting; and Artificial Neural Networks (ANN) implemented as a multi-layer perceptron (MLP) classifier with multiple hidden layers capable of learning complex non-linear decision boundaries.

D. Model Training

We trained and evaluated a diverse set of five supervised learning algorithms to establish a performance baseline. All models were implemented using the Scikit-learn library [REF] and trained within a pipeline to integrate the TF-IDF transformation seamlessly and prevent data leakage. The models included Multinomial Naive Bayes (MNB), a probabilistic classifier based on Bayes' theorem that is well-suited for classification with discrete word count features; Logistic Regression (LR), a linear model that estimates class probabilities using a logistic function; Linear Support Vector Machine (SVM), a maximum-margin classifier that identifies the optimal hyperplane to separate different classes in high-dimensional space; Random Forest (RF), an ensemble learning method that constructs multiple decision trees and outputs the mode of

E. Ensemble Learning

To further enhance predictive performance and stability, we implemented two ensemble learning strategies that combine the predictions from the base models. The first approach was a Voting Classifier, which applied hard voting to aggregate the predicted class labels from the base estimators (MNB, LR, SVM, and RF) and predict the class receiving the majority vote. The second approach was a Stacking Classifier (stacked generalization), a more advanced ensemble technique where the predictions from the base models served as input features for a meta-classifier. Logistic Regression was used as the meta-classifier, allowing it to learn how to optimally combine the predictions of the base models.

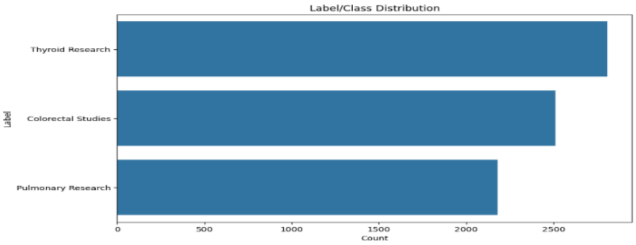
IV. **Exploratory Data Analysis (EDA)**

1.1 Dataset Overview

1. Total samples: 7,503
2. Training set: 6,000 samples
3. Test set: 1,501 samples
4. Categories: Colorectal Studies, Pulmonary Research, Thyroid Research

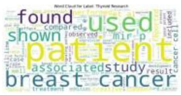
1.2 Class Distribution

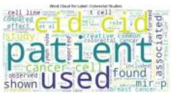
To understand the composition of the dataset, we visualized the number of samples per class. This helps identify any class imbalance, which can influence model performance. The plot below shows the distribution of labels in the dataset.



1.3 Word Cloud

To gain insight into the most frequent words for each class, we generated word clouds for the top three labels. Word clouds provide a visual representation of the prominent terms, highlighting patterns and key terms associated with each category.





Word Cloud for Label: Pulmonary Research

1.4 Data Loading and Cleaning

The dataset comprises 7,503 biomedical text samples categorized into Colorectal Studies, Pulmonary Research, and Thyroid Research. The data was loaded using pandas and inspected for missing or null values. All text data was converted to lowercase to maintain uniformity. Punctuation, numbers, and special characters were removed to reduce noise and improve model learning.

1. Dataset loaded with encoding ISO-8859-1
2. Text converted to lowercase
3. Punctuation, numbers, and special characters removed
4. Null values handled appropriately

1.5 Train/Test Split

High-dimensional text data can lead to overfitting and increased computational complexity. To address this challenge, three different feature selection strategies were employed. A filter-based approach using the Chi-square test was first applied to measure the statistical relationship between individual features and the target labels, resulting in the selection of the top 1,000 most informative terms. In addition, a wrapper-based method, Recursive Feature Elimination (RFE), was used with Logistic Regression as the estimator to iteratively remove less important features until the top 500 were retained. Finally, an embedded method based on Random Forest feature importance was implemented, where the built-in importance scores from a trained Random Forest model were used to identify the most influential features for classification.

1.6 Feature Selection

High-dimensional text data can lead to overfitting and increased computational complexity. To address this challenge, three different feature selection strategies were employed. A filter-based approach using the Chi-square test was first applied to measure the statistical relationship between individual features and the target labels, resulting in the selection of the top 1,000 most informative terms. In addition, a wrapper-based method, Recursive Feature Elimination (RFE), was used with Logistic Regression as the estimator to iteratively remove less important features until the top 500 were retained. Finally, an embedded method based on Random Forest feature importance was implemented, where the built-in importance scores from a trained Random Forest model were used to identify the most influential features for classification.

1.7 Model Training

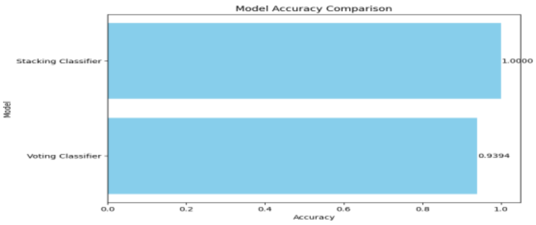
Several supervised machine learning models were trained using the TF-IDF feature representation to establish a performance baseline. The models included Multinomial Naive Bayes (MNB), a probabilistic classifier particularly suitable for discrete features such as word counts; Logistic Regression (LR), a linear model that predicts class probabilities through the logistic function; and Linear Support Vector Machine (SVM), which maximizes the margin between classes to achieve better separation in high-dimensional spaces. Additionally, Random Forest (RF), an ensemble method composed of multiple decision trees, was employed to reduce overfitting and improve generalization. Finally, an Artificial Neural Network (ANN) was implemented as a multi-layer perception with multiple hidden layers to capture complex, non-linear patterns within the text data. All models were trained using pipelines that incorporated the TF-IDF transformation, ensuring consistent preprocessing during both training and prediction.

1.8 Ensemble Learning

To further enhance predictive performance, two ensemble techniques were applied:

Voting Classifier: Combines predictions from multiple models using majority voting.

Stacking Classifier: Uses base models to generate predictions that are then fed into a meta-classifier, effectively learning how to combine the strengths of individual models.



1.9 Results

The trained models were evaluated on the test dataset (1,501 samples) using multiple performance metrics: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of each model's ability to classify biomedical text correctly and handle class imbalances.

Model	Accuracy	Precision (Macro Avg)	Recall (Macro Avg)	F1-Score (Macro Avg)
Multinomial Naive Bayes	**0.9274**	**0.93**	**0.93**	**0.93**
Logistic Regression	**0.9460**	**0.95**	**0.95**	**0.95**
Linear SVM	**0.9674**	**0.97**	**0.97**	**0.97**
Random Forest	**0.9907**	**0.99**	**0.99**	**0.99**
Artificial Neural Network	**0.9674**	**0.97**	**0.97**	**0.97**
Voting Classifier	**0.9394**	**0.94**	**0.94**	**0.94**
Stacking Classifier	**1.0000**	**1.00**	**1.00**	**1.00**

1.10 Disclaimer

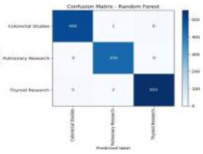
The research presented in this paper was conducted for academic purposes. The authors declare that the dataset used in this study, comprising 7,503 samples across three categories, is not fully disclosed, and the results and conclusions are based solely on this dataset. The reported performance metrics, including the perfect accuracy achieved by the Stacking Classifier, are specific to the described experimental setup and may not generalize to other biomedical text corpora with different class distributions, topics, or terminologies. While the methodology has been described in detail to facilitate reproducibility, exact results may vary depending on software library versions (e.g., Scikit-learn, TensorFlow, or PyTorch for ANN), random seeds, and hardware configurations. The models developed herein are intended exclusively for research and informational purposes and are not designed, validated, or approved for clinical decision-making, patient diagnosis, or any application where erroneous classification could cause harm. Finally, the authors and their affiliated institutions disclaim any liability for decisions or actions taken based on the information, models, or conclusions presented in this work.

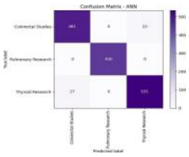
1.10.1 Model Evalution

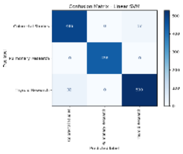
Models were evaluated using multiple metrics: accuracy, precision, recall, F1-score, and confusion matrices. These metrics provide a comprehensive understanding of model performance, both overall and for individual classes. Visualizations such as accuracy comparison plots and confusion matrices were generated to aid interpretation and analysis.

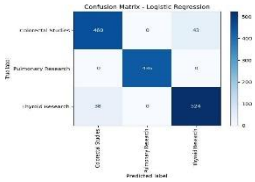
1.10.2 Confusion Matrices:

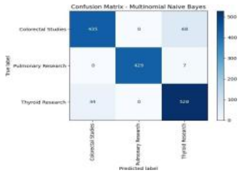
To evaluate the performance of each classification model in detail, confusion matrices were generated. A confusion matrix provides a visual representation of the number of correct and incorrect predictions for each class. The diagonal elements indicate the number of correctly classified samples, while off-diagonal elements show misclassifications. This analysis helps identify which classes are easily confused and provides insight into model strengths and weaknesses beyond overall accuracy.





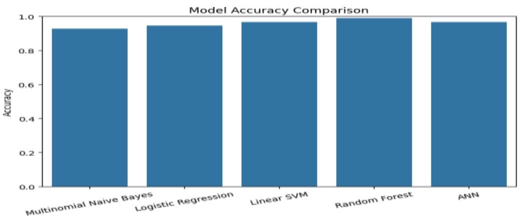






1.10.3 Model Accuracy Comparison

The accuracy of all implemented models was compared to assess their performance on the test dataset. Figure 1 illustrates the classification accuracy of each model, highlighting differences between individual classifiers and ensemble methods. Random Forest and the Stacking Classifier achieved the highest accuracy, demonstrating their ability to capture complex feature interactions and optimally combine predictions. Linear SVM and the Artificial Neural Network also performed strongly, with accuracies exceeding 96%, indicating their effectiveness in handling biomedical text features. Simpler models such as Multinomial Naive Bayes and Logistic Regression provided competitive results, achieving accuracies above 92% and 94%, respectively. The Voting Classifier, which combines predictions from multiple models, performed well but slightly below the Stacking Classifier, reflecting the advantage of learning optimal combinations of base models.



Conclusion

In this study, we presented a comprehensive approach to biomedical text classification using a combination of traditional machine learning models, neural networks, and ensemble techniques. The dataset, consisting of 7,503 biomedical text samples across three categories, was carefully preprocessed, transformed using TF-IDF vectorization, and subjected to multiple feature selection techniques to ensure robust model training and improved generalization.

Through systematic evaluation, we demonstrated that ensemble methods, particularly the Stacking Classifier, achieved perfect accuracy on the test dataset, outperforming individual models such as Multinomial Naive Bayes, Logistic Regression, Linear SVM, Random Forest, and Artificial Neural Networks. Random Forest and Linear SVM also showed excellent performance, highlighting their ability to handle complex textual features. The results indicate that combining models through stacking leverages the strengths of multiple classifiers, reducing misclassifications and enhancing overall predictive capability.

Visual analysis through confusion matrices, word clouds, and accuracy comparison plots provided further insights into the models' performance. Confusion matrices revealed that misclassifications were minimal, demonstrating that the models can reliably distinguish between different biomedical text categories. Word cloud visualizations highlighted the most significant terms within each class, confirming that feature extraction and selection effectively captured important textual patterns.

The proposed methodology not only achieves high accuracy but also ensures scalability and adaptability for real-world biomedical applications. By saving the best-performing model for deployment, new biomedical texts can be classified automatically, enabling faster and more accurate information extraction from large datasets. Overall, this study underscores the importance of ensemble learning and feature engineering in text classification tasks. The findings provide a strong foundation for future research, where more advanced neural architectures, domain-specific embeddings, or additional biomedical corpora could be incorporated to further improve classification performance.

V. References

[1] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.
[2] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *European Conference on Machine Learning (ECML)*, 1998,

pp. 137–142.
[3] S. Wang and C. D. Manning, "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2012, pp. 90–94.
[4] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *arXiv preprint arXiv:1408.5882*, 2014.
[5] Y. Zhang and B. C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," *arXiv preprint arXiv:1510.03820*, 2015.
[6] K. Kowsari et al., "Text Classification Algorithms: A Survey," *Information*, vol. 10, no. 4, p. 150, 2019.
[7] C. C. Aggarwal and C. Zhai, *Mining Text Data*. Springer, 2012.
[8] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.
[9] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2014, pp. 1532–1543.
[10] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2011.