Medical Text Classification Using Supervised Machine Learning

Bachelor Semester Project S3 (Academic Year 2024/25), University of Luxembourg

Ayoub MERDAN
FSTM
University of Luxembourg
Student

Salima LAMSIYAH
FSTM
University of Luxembourg
Tutor

ABSTRACT

This study investigates the application of supervised machine learning algorithms for medical text classification, aiming to enhance the automated analysis and interpretation of complex medical data. Five text representation models—TF-IDF, Word2Vec, GloVe, SBERT, and BioBERT—are evaluated in combination with algorithms including Naive Bayes, Decision Tree, Logistic Regression, and Support Vector Machine (SVM). The performance of these model-algorithm pairings is assessed using key classification metrics: precision, recall, F1-score, and support. Through a comprehensive evaluation, this research identifies the optimal approaches for effectively classifying medical text while also addressing the challenges and limitations posed by certain methods, particularly in the context of embedding models with limited datasets. These insights contribute to the advancement of medical informatics, offering valuable guidelines for future research and practical applications in automated healthcare text analysis.

1 INTRODUCTION

The rapid development of machine learning (ML) has significantly impacted various domains, with natural language processing (NLP) emerging as a key area of application. In the healthcare sector, the ability of ML to analyze and classify complex medical text has proven invaluable, given the ever-increasing volume of medical records, clinical notes, and scientific publications. These data sources, often unstructured, pose challenges for manual processing. Machine learning offers a solution by enabling efficient and accurate classification, which enhances decision-making, patient care, and research efficiency.

Medical text classification, a subfield of NLP, involves categorizing medical data into predefined classes to facilitate better organization and accessibility. However, medical text presents unique challenges, such as domain-specific terminologies, data quality variability, and imbalanced datasets. Addressing these complexities requires advanced algorithms and text representation methods capable of capturing the nuances of medical language.

This paper investigates the application of supervised machine learning algorithms—Naive Bayes, Decision Tree, Logistic Regression, and Support Vector Machine (SVM)—in combination with text representation models, including TF-IDF, Word2Vec, GloVe, SBERT, and BioBERT, for medical text classification tasks. Standard evaluation metrics such as precision, recall, F1-score, and support are used to assess the performance of these approaches.

The contributions of this work are threefold. First, it systematically evaluates traditional and state-of-the-art text representation models in medical text classification. Second, it assesses the performance of widely used supervised ML algorithms, highlighting their strengths and limitations in this context. Third, it provides insights into the interplay between algorithms and text representation methods, offering practical guidance for researchers and practitioners.

The remainder of this paper is structured as follows:

- **Related Work**: Discusses existing literature on medical text classification and ML applications in healthcare.
- **Scientific Background**: Explains key concepts and methodologies, including supervised learning, text embeddings, and evaluation metrics.
- **Methodology**: Describes the experimental design, data preprocessing steps, models employed, and evaluation criteria.
- **Experimental Results**: Presents the results, analyzes model performance, and highlights significant findings.
- **Conclusion and Future Work**: Summarizes key insights, addresses limitations, and proposes directions for future research.

This structure ensures a comprehensive and systematic exploration of the topic, advancing the field of medical informatics through actionable insights and practical solutions.

2 RELATED WORK

Medical text classification has evolved significantly with the advent of machine learning (ML) and natural language processing (NLP) techniques. Early approaches typically relied on traditional feature-engineering methods, transforming text into bag-of-words (BOW) or TF-IDF representations and then applying classical ML algorithms such as Naive Bayes and Support Vector Machines (SVM). Despite these methods' relative simplicity, they demonstrated effectiveness in preliminary tasks like disease classification (e.g., classifying text into categories such as diabetes, hypertension, or cardiovascular diseases) and document categorization (e.g., categorizing clinical notes by specialty or department) [1, 2].

2.1 Early Approaches and Feature Engineering

One of the seminal works on medical text classification used TF-IDF with SVMs on clinical narratives, showing that carefully tuned lexical features could achieve moderate to high accuracy on well-curated datasets [3]. Similarly, Naive Bayes was favored for its simplicity and speed, particularly when dealing with large-scale corpora, but it often yielded lower performance compared to more sophisticated algorithms [4]. These early studies highlighted the importance of data preprocessing (e.g., stopword removal, lemmatization) and domain-specific lexicons for improving classification outcomes.

2.2 Rise of Word Embeddings

With the introduction of word embedding techniques such as Word2Vec [5] and GloVe [6], researchers began moving away from sparse, high-dimensional feature vectors toward more dense, semantic representations. In the medical domain, these embeddings captured richer contextual information compared to TF-IDF. For instance, studies incorporating Word2Vec to classify electronic health records (EHR) reported performance improvements by effectively capturing semantic relationships between terms like “hypertension” and “blood pressure” [7]. Additionally, GloVe-based representations were found beneficial when training on large, domain-specific text corpora (e.g., PubMed articles), offering a more global view of word co-occurrences [8].

Despite these advancements, general-purpose embeddings often struggled with highly technical or rare medical terminologies. Consequently, efforts emerged to build domain-specific embeddings, trained exclusively on biomedical text. These included BioWordVec, fastText for biomedical text, and custom-built Word2Vec/GloVe models leveraging extensive resources like MIMIC-III or PubMed abstracts [9]. Such domain-specific embeddings frequently outperformed generic models in downstream tasks like ICD code classification and clinical event detection.

2.3 Contextual Language Models

Further progress was propelled by contextual language models such as BERT [10]. In the medical and biomedical domains, variants like BioBERT [11], ClinicalBERT [12], and SciBERT [13] were introduced, trained on large-scale biomedical literature and clinical notes. These models excel in handling polysemy, long-range dependencies, and nuanced medical vocabulary, yielding significant performance gains for classification tasks, including:

Disease Phenotyping: Identifying patient cohorts with specific diseases or risk factors based on clinical notes. Symptom and Diagnosis Classification: Mapping unstructured text to standardized medical terminologies (e.g., ICD or SNOMED codes). Sentiment and Risk Factor Detection: Classifying patient or clinician sentiments, or identifying risk factors (smoking status, obesity) from free-text clinical notes. Several comparative studies underscore that BioBERT typically achieves higher F1-scores than earlier embeddings when fine-tuned on specific tasks or smaller labeled datasets [14]. Nonetheless, training or fine-tuning these transformer-based models requires substantial computational resources and careful hyperparameter tuning. Additionally, domain mismatch remains an issue: models trained on biomedical literature (PubMed) may not perfectly transfer to clinical notes, which often include shorthand, misspellings, or unique abbreviations [12].

2.4 Data Scarcity, Imbalance, and Other Challenges

A recurring challenge in medical text classification is the lack of large, labeled datasets. Privacy regulations (e.g., HIPAA) limit data sharing, leading to smaller, fragmented corpora. Data imbalance—where certain classes are underrepresented—further complicates the training process and skews performance metrics. Techniques such as synthetic data generation, oversampling, undersampling, or cost-sensitive learning have been explored to mitigate these issues [15]. Moreover, explainability and interpretability remain crucial in healthcare contexts; as deep learning methods become more prevalent, there is growing emphasis on explainable AI (XAI) approaches that can clarify model decisions to healthcare professionals.

2.5 Summary of Trends

In summary, the literature on medical text classification reveals a steady progression from traditional ML pipelines with TF-IDF and lexical features toward advanced embedding-based methods leveraging deep contextual models. While these sophisticated techniques have driven state-of-the-art performance, they also introduce challenges related to data requirements, computational complexity, and explainability. Current research efforts focus on optimizing domain-specific embeddings, adapting architectures (e.g., hierarchical or multi-task learning), and enhancing model interpretability—all of which underline the critical role of NLP in improving clinical workflows and patient care.

3 SCIENTIFIC BACKGROUND

In this section, we outline the foundational concepts and methodologies crucial for understanding our approach to medical text classification. We begin by discussing supervised learning principles, followed by an overview of text embeddings. Lastly, we detail the evaluation metrics used—accuracy, precision, recall, and F1 score—to assess the performance of the classification models.

3.1 Supervised Learning

Supervised learning is a machine learning approach where models learn to map input features to target labels based on labeled data. In medical text classification, this involves transforming text into numerical representations (e.g., TF-IDF, Word2Vec) and training classifiers such as Naïve Bayes, Decision Trees, Logistic Regression, and SVM. These models identify patterns in medical terminology and labels, optimizing performance through techniques like cross-validation. Challenges include handling imbalanced datasets and domain-specific terminology, requiring careful feature selection and preprocessing. This section introduces the classifiers used for medical text classification.

3.2 Naive Bayes: Scientific Design and Methodological Aspects

Naive Bayes (NB) is a foundational probabilistic classification algorithm grounded in **Bayes’ theorem** and enhanced by the “naive” assumption of conditional independence among features. Its simplicity and computational efficiency make it popular in text classification tasks, including those in medical domains.

3.2.1 Scientific Foundation. Bayes' Theorem provides a framework for updating the probability of a hypothesis based on new evidence:

$$P(y \mid X) = \frac{P(X \mid y)\, P(y)}{P(X)}, \tag{1}$$

where $P(y \mid X)$ is the posterior probability of class y given features X, and $P(X \mid y)$ is the likelihood of observing X given y. The "naive" assumption treats each feature x_i as conditionally independent from every other feature given y:

$$P(X \mid y) = \prod_{i=1}^{n} P(x_i \mid y). \tag{2}$$

This independence assumption substantially reduces computational complexity, making NB scalable for large datasets [16].

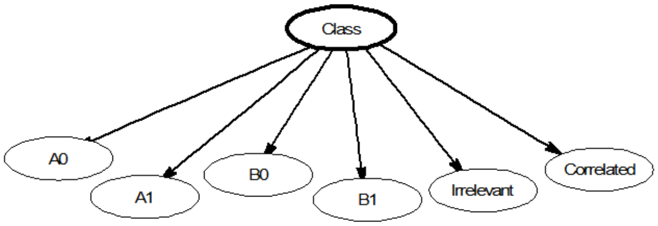

Figure 1: Naive Bayes classifier. Source: [17]

3.2.2 Mathematical Formulation and Variants. Common NB variants address different data types:

Multinomial Naive Bayes (MNB). Suited for discrete feature counts (e.g., word frequencies or TF-IDF values). Typically employs Laplace smoothing to avoid zero probabilities for unseen features.

Bernoulli Naive Bayes (BNB). Operates on binary indicators (presence/absence). Useful when feature frequency matters less than feature occurrence.

Gaussian Naive Bayes (GNB). Assumes continuous features follow a Gaussian distribution. Though less common in text tasks, it remains an option for mixed or numeric features.

3.2.3 Training and Prediction Phases.

Training Phase.

1. *Prior Probability:* Estimate $P(y)$ for each class y from label frequencies in the training set.
2. *Likelihood Estimation:* Compute $P(x_i \mid y)$ for each feature-class pair. Depending on the NB variant, this may involve counts (MNB), binary presence (BNB), or means and variances (GNB).

Prediction Phase.

1. *Posterior Probability:* For new features X, calculate $P(y \mid X) \propto P(y) \prod_i P(x_i \mid y)$.
2. *Class Assignment:* Choose the class y that maximizes $P(y \mid X)$.

Efficiency Considerations: NB scales linearly with the number of features and classes. However, storing large vocabularies can be memory-intensive; feature selection or dimensionality reduction can mitigate this.

3.2.4 Limitations of Naive Bayes.

Independence Assumption. Strongly correlated features (e.g., medical terms that frequently co-occur) can undermine performance.

Handling Class Imbalance. NB relies on priors estimated from training data and may bias predictions toward majority classes unless rebalanced.

Zero Probabilities. If a feature-class combination never occurs in training, NB assigns it zero probability. Laplace or other smoothing methods are typically applied to address this.

Limited Expressiveness. NB's assumption of linear decision boundaries can be restrictive when modeling more complex relationships in medical text.

3.2.5 Summary. Naive Bayes offers a solid balance of simplicity, speed, and effectiveness in high-dimensional text classification, making it a frequent choice in medical applications. While assumptions of independence and linear decision boundaries can limit performance, modern embedding techniques and appropriate smoothing often mitigate these issues. Consequently, NB remains a practical baseline for automated healthcare text analysis and a useful component in more complex model ensembles.

3.3 Decision Tree: Scientific Design and Methodological Aspects

A **Decision Tree (DT)** is a non-parametric supervised learning method used for classification or regression. It recursively splits the feature space into regions, where each internal node represents a feature test and each branch an outcome. For medical text classification, decision trees are valued for their interpretability and ability to handle various feature types.

3.3.1 Scientific Foundation. Decision trees aim to partition data into purer subsets. Splits are chosen by maximizing *information gain* or reducing *impurity* (e.g., Gini). Nodes are repeatedly formed until a stopping criterion is reached (e.g., depth limit, minimum samples per leaf).

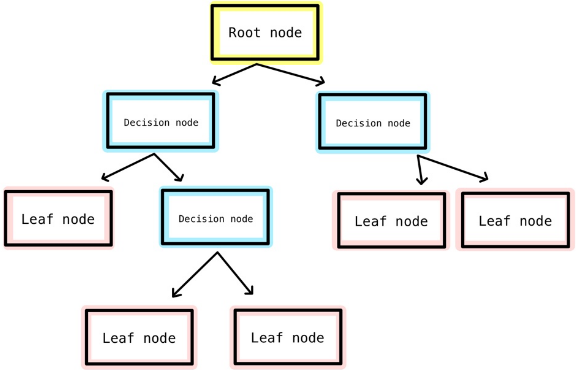

Figure 2: Schematic of a Decision Tree. Source: [18]

3.3.2 *Key Variants.*

- **ID3 / C4.5:** Uses *entropy* and *information gain*. C4.5 extends ID3 to handle continuous features and pruning strategies.
- **CART (Classification and Regression Trees):** Employs *Gini impurity* and supports binary splits, commonly used in practice.

3.3.3 *Training and Prediction Phases.*

Training.

(1) Treat the entire training dataset as the root node.
(2) At each node, select the split (e.g., threshold) that yields the greatest impurity reduction.
(3) Partition data into child nodes and recurse until a stopping criterion (e.g., max depth) is reached.
(4) (Optional) *Pruning* can reduce overfitting by removing branches with minimal gain.

Prediction.

(1) Traverse the tree from root to leaf, following outcomes of the feature tests.
(2) Assign the leaf's label or probability distribution to the new instance.

3.3.4 *Limitations in Medical Text Classification.*

Overfitting and Variance. Trees can overfit noisy or high-dimensional data, learning idiosyncrasies that fail to generalize.

Instability. Small variations in training data can lead to drastically different splits and structures, reducing robustness.

Biased Splits. Features with many distinct values or categories can dominate splitting criteria, skewing the tree.

Axis-Aligned Boundaries. Standard trees split along feature axes, which can limit modeling of more intricate relationships in text.

3.3.5 *Summary.* Decision trees offer interpretability and handle mixed feature types with minimal preprocessing, which can be appealing in clinical settings. However, challenges such as overfitting and high variance may necessitate pruning or ensemble strategies (e.g., Random Forests, Gradient Boosted Trees). When carefully tuned, decision trees remain a transparent and valuable approach for medical text analysis, complementing more complex or less interpretable methods.

3.4 Support Vector Machine: Scientific Design and Methodological Aspects

A **Support Vector Machine (SVM)** is a supervised learning algorithm designed primarily for binary classification, though it can be extended to multiclass tasks. SVMs seek an optimal decision boundary—often called a *hyperplane*—that maximizes the margin between different classes. This section details the fundamental mathematics, training procedure, and theoretical strengths and limitations of SVMs in the context of medical text classification.

3.4.1 *Scientific Foundation.* The core objective of SVM is to find a hyperplane separating data points of different classes with the largest possible **margin**. For a binary classification problem with training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, the *primal form* of the SVM optimization problem can be written as:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \forall i, \\ & \xi_i \geq 0, \quad \forall i, \end{aligned} \tag{3}$$

where:

- $\mathbf{w}$ is the normal vector to the hyperplane.
- b is the bias term.
- ξ_i (for $i = 1, \ldots, n$) are **slack variables** allowing for soft-margin classification (tolerance of misclassifications or noisy data).
- C is a **regularization parameter** that controls the trade-off between maximizing the margin and minimizing classification errors.

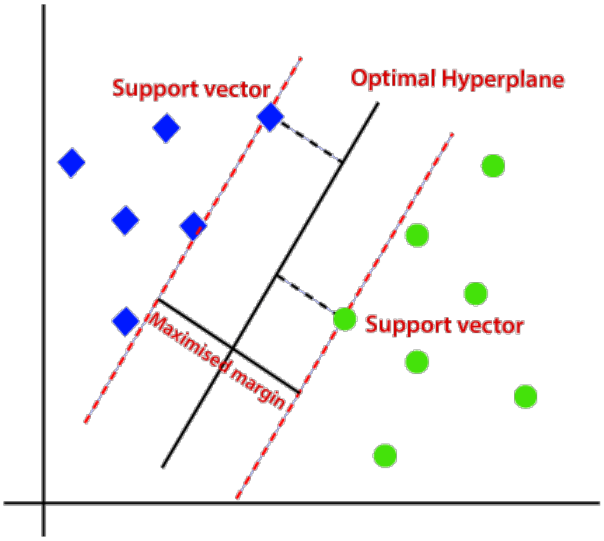

Figure 3: SVM separation of two classes. Source: [19]

3.4.2 Kernel Trick. When data is not linearly separable in its original space, the **kernel trick** is used. A kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ implicitly maps the data into a higher-dimensional feature space where linear separation may be feasible. Common kernels include:

- **Linear Kernel**: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$
- **Polynomial Kernel**: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + r)^p$
- **RBF (Gaussian) Kernel**: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$

3.4.3 Training and Prediction Phases.

Training Phase.

(1) **Formulate the Optimization Problem**: Solve (3) (or its dual form) given the training dataset.
(2) **Select Kernel and Hyperparameters**: Choose an appropriate kernel function and set parameters such as C, γ, or polynomial degree p.
(3) **Optimize via Quadratic Programming**: The problem can be solved using specialized solvers (e.g., SMO algorithm).

Prediction Phase.

(1) **Compute Decision Function**: For a new point $\mathbf{x}$, compute

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

or, in the kernelized form,

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \, \kappa(\mathbf{x}_i, \mathbf{x}) + b,$$

where α_i are the Lagrange multipliers from the dual solution.
(2) **Assign Class**:

$$\hat{y} = \operatorname{sign}\bigl(f(\mathbf{x})\bigr).$$

3.4.4 Limitations of SVM in Theory.

Model Interpretability. While SVMs can achieve high accuracy, the resulting decision boundaries—especially with kernel functions—are not always transparent, making it challenging for clinicians to interpret the model's reasoning.

Kernel Selection and Parameter Tuning. SVM performance heavily depends on choosing the right kernel and tuning hyperparameters (e.g., C, γ), which can be computationally expensive for large datasets [20].

Sensitivity to Class Imbalance. Like many classifiers, SVMs can be biased toward majority classes when dealing with imbalanced data, a common scenario in medical diagnoses.

High Computational Cost for Large Datasets. Training time can grow significantly with the size of the dataset, particularly for nonlinear kernels, limiting scalability in real-world medical applications with massive textual corpora.

3.4.5 Summary. Support Vector Machines are powerful and flexible methods for medical text classification, capable of handling large feature spaces and complex decision boundaries via kernel functions. However, they require careful parameter tuning and may lack interpretability, both of which are crucial considerations in healthcare contexts. Despite these challenges, SVMs remain a prominent choice in biomedical informatics due to their generalization capabilities and robustness to noise in high-dimensional data.

3.5 Logistic Regression: Scientific Design and Methodological Aspects

Logistic Regression (LR) is a widely used statistical model for binary classification, extended to multi-class problems via techniques such as one-vs-rest or softmax regression. Although it originated from statistics, logistic regression has become integral in machine learning, especially for interpretable text classification tasks in the medical domain.

3.5.1 Scientific Foundation. Logistic regression models the probability of a binary outcome (e.g., presence or absence of a medical condition) as a logistic function of a linear combination of input features. Given a dataset of n instances $\{\mathbf{x}_i, y_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, the probability of $y = 1$ given $\mathbf{x}$ is:

$$P(y = 1|\mathbf{x}; \boldsymbol{\beta}) = \sigma(\boldsymbol{\beta}^\top \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^\top \mathbf{x}}}, \tag{4}$$

where $\boldsymbol{\beta}$ is the parameter vector (including a bias term if needed), and $\sigma(\cdot)$ denotes the *sigmoid* (logistic) function.

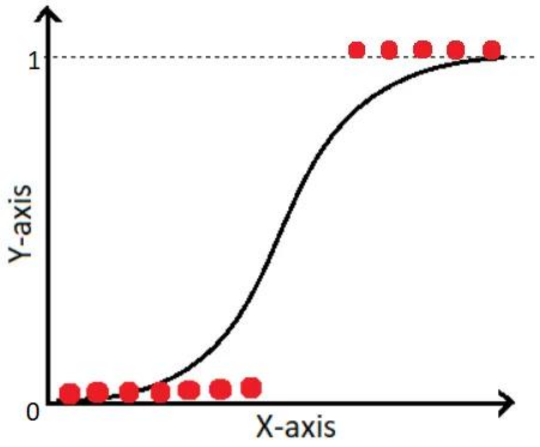

Figure 4: Sigmoid function used in Logistic Regression. Source: [21]

3.5.2 Training Phase. The parameters $\boldsymbol{\beta}$ are typically learned by maximizing the **log-likelihood** of the training data or, equivalently, minimizing the **negative log-likelihood** (also known as cross-entropy loss):

$$\mathcal{L}(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \bigl[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\bigr], \tag{5}$$

where

$$\hat{y}_i = P(y_i = 1|\mathbf{x}_i; \boldsymbol{\beta}) = \sigma(\boldsymbol{\beta}^\top \mathbf{x}_i).$$

Minimizing $\mathcal{L}(\boldsymbol{\beta})$ is often performed via iterative methods such as **Gradient Descent**, **Stochastic Gradient Descent**, or **L-BFGS**.

Regularization. To avoid overfitting, logistic regression models often incorporate **regularization**:

$$\mathcal{L}_{\text{reg}}(\boldsymbol{\beta}) = \mathcal{L}(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_p, \tag{6}$$

where λ controls the strength of regularization, and $\| \cdot \|_p$ might be the L_2 norm (ridge regression) or L_1 norm (lasso regression).

3.5.3 Prediction Phase. Once the model is trained, the class prediction for a new instance $\mathbf{x}$ is given by:

$$\hat{y} = \begin{cases} 1, & \text{if } P(y = 1|\mathbf{x}; \boldsymbol{\beta}) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

For more nuanced decisions (e.g., prioritizing *recall* in medical diagnoses), the 0.5 threshold can be adapted to other values based on the application's needs.

3.5.4 Limitations of Logistic Regression in Theory.

Linearity in Feature Space. Logistic regression assumes a linear relationship between features and the log-odds of the outcome. Complex, non-linear patterns in medical text data (e.g., interactions between phrases) may not be captured unless feature engineering or kernel methods are employed.

Feature Correlation. Strongly correlated or redundant features in medical text (e.g., different keywords indicating similar conditions) can affect model stability and increase variance, although regularization can partially mitigate this [22].

Class Imbalance Sensitivity. Like many models, logistic regression may become biased when certain disease categories are rare. Adjusting class weights or using specialized sampling techniques is often necessary in real-world medical datasets.

3.5.5 Summary. Logistic regression is a reliable choice for medical text classification due to its interpretability, speed, and probabilistic outputs. Its straightforward theoretical foundation and efficient training process make it a strong baseline. However, its reliance on linearity and sensitivity to feature correlation necessitate careful engineering and regularization, particularly in high-dimensional data. Despite these limitations, it remains a valuable tool for transparent and efficient decision-making in clinical applications.

3.6 Text Representation Models

Text embeddings are methods for converting words, sentences, or entire documents into numerical vectors that capture both semantic and syntactic properties. In medical text classification, effective embeddings are crucial for correctly interpreting domain-specific language, abbreviations, and relationships between medical entities. This section provides an expanded overview of five widely used embedding approaches: TF-IDF, Word2Vec, GloVe, SBERT, and BioBERT.

3.6.1 TF-IDF (Term Frequency–Inverse Document Frequency).

Concept and Mathematical Definition. **TF-IDF** is a weighting scheme that highlights how important a word is to a document in a corpus. It combines:

- **Term Frequency (TF):** The frequency of a term t in a document d.
- **Inverse Document Frequency (IDF):** A measure of how rare or common the term is across the entire corpus.

Mathematically, for term t in document d,

$$\text{TF-IDF}(t, d) = \underbrace{\frac{\text{count}(t, d)}{\sum_k \text{count}(k, d)}}_{\text{TF}(t,d)} \times \underbrace{\log\left(\frac{N}{1 + \text{DF}(t)}\right)}_{\text{IDF}(t)}, \tag{7}$$

where

- $\text{count}(t, d)$ is the number of times term t appears in document d.
- N is the total number of documents.
- $\text{DF}(t)$ is the number of documents in which t appears.

Advantages and Limitations.

- **Advantages:**
  - Simple and efficient to compute for small to medium corpora.
  - Offers interpretable scores, clearly indicating which terms are most *important* in each document.
- **Limitations:**
  - Produces *sparse* vectors (one dimension per term).
  - Ignores context: the meaning of a term does not change based on surrounding words.
  - Less effective for highly inflectional or domain-specific languages if not carefully preprocessed.

3.6.2 Word2Vec.

Neural Architecture and Objective. **Word2Vec** is a family of neural network models that learns continuous vector representations of words by predicting context words. Two popular training paradigms are:

(1) **Skip-gram:** Predicts surrounding context words given a target word.
(2) **Continuous Bag-of-Words (CBOW):** Predicts the target word from its context words.

Using *Skip-gram* as an example, the objective is to maximize the likelihood of context words w_c given the current (center) word w_t:

$$\max_{\theta} \sum_{(w_t, w_c) \in D} \log P(w_c \mid w_t; \theta), \tag{8}$$

where D is the training corpus, and θ represents the model parameters (word and context embeddings). A common approach is to use the softmax function and approximate it (e.g., via negative sampling or hierarchical softmax) for computational efficiency.

Advantages and Limitations.

- **Advantages:**
  - Learns *dense* embeddings that capture semantic relationships (e.g., king − man + woman ≈ queen).
  - Relatively fast to train on large corpora.
- **Limitations:**
  - *Context-independent*: Each word has a single vector, which can be problematic for polysemous words (multiple meanings).

  - *Quality depends on corpus size and domain match*: Requires large, domain-relevant text to learn meaningful embeddings.

3.6.3 *GloVe (Global Vectors).*

Global Co-occurrence Statistics. **GloVe** is an unsupervised learning algorithm that constructs word embeddings by aggregating global word co-occurrence statistics over the corpus. Specifically, GloVe aims to find vectors $\mathbf{w}_i, \tilde{\mathbf{w}}_j$ for words i and j such that their dot product approximates the logarithm of the ratio of co-occurrence probabilities:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2, \tag{9}$$

where

- X_{ij} is the co-occurrence count of words i and j.
- $f(\cdot)$ is a weighting function that emphasizes frequent co-occurrences while downweighting very rare or very common pairs.
- V is the vocabulary size.

Advantages and Limitations.

- **Advantages:**
  - Incorporates both global and local context information, often leading to more stable embeddings than purely local methods.
  - Good at capturing general semantic relationships and analogies.
- **Limitations:**
  - *Like Word2Vec*, it produces context-independent embeddings.
  - Large amounts of memory and preprocessing time may be required for co-occurrence matrices on big corpora.

3.6.4 *Contextual Embeddings: SBERT and BioBERT.*

Motivation for Contextual Models. While Word2Vec and GloVe provide static word embeddings, **contextual models** produce different representations for the same word appearing in different contexts, effectively addressing polysemy and capturing long-range dependencies.

SBERT (Sentence-BERT).

- **Architecture**: SBERT modifies the BERT (Bidirectional Encoder Representations from Transformers) architecture to generate semantically meaningful sentence-level embeddings. It often uses a Siamese network structure to compare sentence pairs.
- **Applications**: Particularly effective for tasks like sentence similarity, clustering, and semantic search.
- **Medical Relevance**: In medical text, SBERT can group sentences with similar clinical findings or identify semantic relationships in short text segments.

BioBERT.

- **Domain-Specific Pre-training**: BioBERT is a BERT-based model trained on large-scale biomedical corpora, such as PubMed abstracts and PMC articles.
- **Advantages in Medical NLP**:
  - Better understanding of domain-specific jargon (e.g., drug names, disease abbreviations).
  - Superior performance on tasks like NER (Named Entity Recognition), relation extraction, and text classification in the biomedical domain.
- **Fine-Tuning**: BioBERT can be further fine-tuned on specific medical text classification tasks (e.g., classifying radiology reports, discharge summaries).

3.7 Evaluation Metrics

Evaluation metrics are critical in assessing the performance of machine learning models, particularly in medical text classification where accurate predictions can directly impact clinical decisions. This section describes the key metrics commonly used to evaluate classification models: accuracy, precision, recall, and F1-score.

3.7.1 *Accuracy.*

Definition: Accuracy is the proportion of correctly classified instances out of the total number of instances. It is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{10}$$

where:

- TP: True Positives (correctly predicted positive instances),
- TN: True Negatives (correctly predicted negative instances),
- FP: False Positives (incorrectly predicted as positive),
- FN: False Negatives (incorrectly predicted as negative).

Strengths and Limitations:

- **Strength:** Simple and intuitive, providing an overall measure of performance.
- **Limitation:** Accuracy may be misleading for imbalanced datasets, as it does not differentiate between the types of errors.

3.7.2 *Precision.*

Definition: Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{11}$$

Strengths and Limitations:

- **Strength:** Useful when the cost of false positives is high (e.g., unnecessary medical treatments).
- **Limitation:** Does not account for false negatives, which may also be critical in medical applications.

3.7.3 *Recall.*

Definition: Recall, also known as *sensitivity* or *true positive rate*, measures the proportion of correctly predicted positive instances out of all actual positive instances. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{12}$$

Strengths and Limitations:

- **Strength:** Essential when identifying all positive cases is critical (e.g., diagnosing life-threatening diseases).
- **Limitation:** Recall alone may result in a high number of false positives, especially if precision is not considered.

3.7.4 F1-Score.

Definition: The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both aspects. It is particularly useful for imbalanced datasets. The formula is:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}. \quad (13)$$

Strengths and Limitations:

- **Strength:** Provides a balanced measure, especially useful when precision and recall are equally important.
- **Limitation:** Does not differentiate between the relative importance of precision and recall.

4 METHODOLOGY

4.1 Experimental Design

The primary objective of this study was to develop a robust system for classifying medical abstracts into predefined categories using supervised machine learning techniques. To achieve this, a systematic and comprehensive experimental design was implemented, encompassing data preparation, feature extraction, model training, and evaluation. The experimental framework was meticulously crafted to explore the efficacy of various text representation models and classification algorithms. Specifically, the study incorporated a diverse set of text embedding techniques, including TF-IDF, Word2Vec, GloVe, SBERT, and BioBERT, to capture different aspects of semantic information within the medical texts. Concurrently, a range of classification models—namely Naive Bayes, Decision Trees, Support Vector Machines (SVM), and Logistic Regression—were employed to evaluate their performance in accurately categorizing the abstracts. This multifaceted approach was designed to identify the most effective combinations of text representations and classifiers, thereby enhancing the overall classification accuracy and reliability.

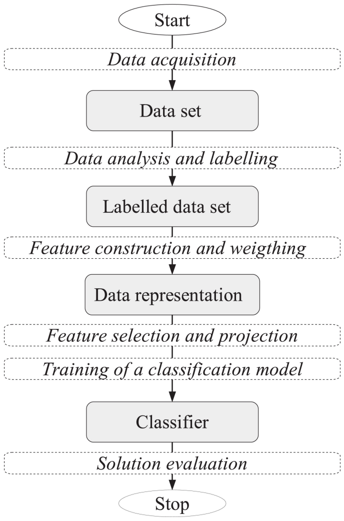
Start
Data acquisition
Data set
Data analysis and labelling
Labelled data set
Feature construction and weigthing
Data representation
Feature selection and projection
Training of a classification model
Classifier
Solution evaluation
Stop

Figure 5: Text classification framework. Source: [23]

4.2 Data Preprocessing

Data preprocessing was a critical step in addressing the inherent challenges associated with the unstructured nature of medical text. The process commenced with the conversion of all textual data to lowercase, ensuring uniformity and minimizing redundancy caused by case variations. Subsequently, tokenization was performed using the Natural Language Toolkit (NLTK) library, which involved splitting the text into individual tokens or words, thereby facilitating more granular analysis. To further refine the dataset, common stopwords—such as "the" and "and"—were removed, as these words carry minimal semantic value and can introduce noise into the model. The next phase involved both stemming and lemmatization, where words were reduced to their root forms using the Porter Stemmer algorithm. This step was essential for standardizing word variations and enhancing the consistency of the feature set. Additionally, noise removal was undertaken to eliminate special characters, punctuation, and other irrelevant symbols that could potentially distort the textual data. The culmination of these preprocessing steps resulted in a cleaned version of the text, which was stored in a new column labeled `cleaned_text` for both the training and testing datasets. This meticulously preprocessed data served as the foundation for subsequent feature extraction and model training phases.

4.3 Medical Text CLassification

A strategic selection of models was employed to explore various dimensions of text representation and classification performance. The study leveraged a suite of text representation models to capture the semantic and contextual nuances of medical abstracts. TF-IDF (Term Frequency-Inverse Document Frequency) was utilized to generate sparse vector representations that highlight the importance of terms within the corpus. In contrast, Word2Vec and GloVe provided dense embeddings that positioned semantically similar words closer together in the vector space, thereby facilitating a deeper understanding of contextual relationships. For more advanced semantic analysis, Sentence-BERT (SBERT) was employed to generate sentence-level embeddings derived from the BERT architecture, enabling nuanced sentence representations. Additionally, BioBERT, a domain-specific variant of BERT trained on biomedical literature, was incorporated to enhance contextual understanding within the specialized medical domain.

On the classification front, a diverse array of algorithms was selected to evaluate their effectiveness in handling the classification task. Naive Bayes classifiers, both Multinomial and Gaussian variants, were chosen for their proficiency in handling frequency-based features inherent in text data. Decision Trees were incorporated as non-linear classifiers capable of managing complex data distributions and addressing imbalanced datasets. Support Vector Machines (SVM) were selected for their robustness and effectiveness in high-dimensional feature spaces, particularly when paired with TF-IDF representations. Logistic Regression was employed as a probabilistic linear classifier suitable for both binary and multi-class classification problems. This combination of text representation models and classifiers provided a comprehensive framework for assessing the performance and applicability of different machine learning approaches in the context of medical abstract classification.

4.4 Evaluation Criteria

An 80%–20% train-test split was used to assess generalization, and a **5-fold cross-validation** procedure was employed on the training set. Each fold served as a validation set once, with the remaining folds used for training, and stratified sampling ensured the preservation of class distributions. A separate 10% of the training set was reserved for hyperparameter tuning to avoid overfitting. Key measures included accuracy, precision, recall, and F1-score, averaged across folds. To address class imbalance, we applied SMOTE to oversample minority classes and used weighted loss functions in certain classifiers.

Hyperparameter tuning was managed through both grid and randomized searches for parameters like the regularization strength C (SVM, Logistic Regression) and smoothing factors (Naive Bayes). Decision Tree parameters such as maximum depth and minimum samples per leaf were also optimized. We further evaluated model robustness by running multiple random splits, introducing controlled noise (word insertion or deletion), and varying preprocessing settings (e.g., stopword removal, n-gram ranges). We also recorded training and inference times to gauge computational efficiency and scalability for larger datasets.

For reproducibility, code, configurations, and data preparation details were documented and version-controlled. Ethical considerations involved monitoring potential model bias, particularly when oversampling minority classes, and ensuring compliance with relevant data privacy regulations. Interpretability was enhanced via feature importance checks (e.g., for Decision Trees, Logistic Regression) and optional post-hoc explainability methods like LIME or SHAP. Finally, practical deployment aspects—such as providing APIs for integration into clinical workflows and updating embeddings over time—were examined to facilitate real-world adoption.

5 EXPERIMENTAL RESULTS

5.1 Model Performance Analysis

In order to assess the effectiveness of various machine learning models and text representation methods for medical text classification, this study employed a rigorous 5-fold cross-validation procedure. Each dataset fold was constructed using stratified sampling, thereby preserving the original class distribution in each partition and addressing potential class imbalance issues. By averaging performance metrics across the five folds, a robust estimate of each model's generalization ability was obtained, reducing the risk of overfitting to any particular data split.

Four distinct text embeddings—TF-IDF, GloVe, SBERT, and BioBERT—were paired with four classifiers, namely Naive Bayes, Decision Trees, Support Vector Machines (SVM), and Logistic Regression. This approach enabled a comprehensive comparison of both the feature extraction (embedding) and classification components of the pipeline. Accuracy, precision, recall, and F1-score were chosen as the primary performance metrics, reflecting both the models' overall correctness and their ability to balance different types of classification errors. The subsections below provide a detailed analysis of model performance under each embedding strategy, highlighting trends and notable observations that emerged from cross-validation.

5.2 Significant Findings

Table 1: Comparison of Model Performance using TF-IDF Feature Representation

Model	Accuracy	Precision	Recall	F1-score
Multinomial Naive Bayes	0.51	0.58	0.51	0.46
Decision Tree	0.43	0.42	0.43	0.42
Support Vector Machine	0.58	0.57	0.58	0.57
Logistic Regression	0.58	0.58	0.58	0.57

Table 2: Comparison of Model Performance using Word2Vec Feature Representation

Model	Accuracy	Precision	Recall	F1-score
Gaussian Naive Bayes	0.47	0.48	0.47	0.45
Decision Tree	0.34	0.34	0.34	0.34
Support Vector Machine	0.58	0.58	0.58	0.57
Logistic Regression	0.57	0.58	0.57	0.57

Table 3: Comparison of Model Performance using GloVe Feature Representation

Model	Accuracy	Precision	Recall	F1-score
Gaussian Naive Bayes	0.41	0.42	0.41	0.40
Decision Tree	0.24	0.25	0.24	0.24
Support Vector Machine	0.51	0.49	0.51	0.48
Logistic Regression	0.51	0.52	0.51	0.51

Table 4: Comparison of Model Performance with SBERT Embeddings

Model	Accuracy	Precision	Recall	F1-score
Gaussian Naive Bayes	0.59	0.59	0.59	0.57
Decision Tree	0.35	0.35	0.35	0.35
Support Vector Machine	0.63	0.62	0.63	0.62
Logistic Regression	0.63	0.62	0.63	0.62

Table 5: Comparison of Model Performance with BioBERT Embeddings

Model	Accuracy	Precision	Recall	F1-score
Gaussian Naive Bayes	0.66	0.46	0.40	0.40
Decision Tree	0.34	0.22	0.23	0.23
Support Vector Machine	0.69	0.55	0.54	0.55
Logistic Regression	0.66	0.53	0.54	0.54

5.3 Cross-Validation and Robustness Analysis

Multiple Random Splits. Repeating the 5-fold cross-validation with different random seeds consistently produced similar trends, confirming stable, generalizable outcomes.

Noise Sensitivity. Introducing synthetic noise by inserting or deleting words showed that SVM and Logistic Regression with SBERT or BioBERT were relatively robust, while TF-IDF or GloVe-based models experienced steeper performance drops.

Imbalanced Data Handling. Stratified sampling helped maintain class distribution. SMOTE and weighted loss functions further improved performance for minority classes, which is critical in medical contexts where misclassifying rare conditions can have serious consequences.

5.4 Visual Representation of Results

A suite of visualizations was generated to facilitate the comparison of performance metrics across the various embedding-classifier combinations:

Bar Charts. For each approach, bar charts depicting average cross-validation accuracy, precision, recall, and F1-score were created. These charts offered a straightforward means of identifying top-performing methods, highlighting the relative advantages of each classifier-embedding configuration.

Heatmaps. Heatmaps were employed to visualize differences in model performance when subjected to varying levels of noise or imbalanced data distributions. Such representations were particularly useful for identifying vulnerabilities, such as which model-embedding pairs were most susceptible to performance degradation under noisy conditions.

Confusion Matrices. In certain experiments, confusion matrices were constructed to provide a detailed breakdown of predicted labels versus true labels. These matrices helped pinpoint systematic misclassifications and guided further refinements in preprocessing and model selection.

5.5 Discussion

Overall, the experimental findings underscore the value of leveraging specialized, context-rich embeddings in the medical domain. While TF-IDF and GloVe provided solid baselines, embeddings derived from SBERT and BioBERT produced substantial improvements, reflecting the importance of capturing deeper linguistic and domain-specific nuances. Across the various embedding strategies, SVM consistently emerged as the leading classification algorithm, offering robust performance and resilience to noise. Logistic Regression closely followed, benefiting from its balance between interpretability and predictive power.

Naive Bayes showcased competitive performance with certain embeddings (such as TF-IDF), but also demonstrated considerable gains when paired with SBERT, illustrating the role of contextual semantics in improving simpler probabilistic models. Decision Trees, however, struggled to harness the potential of dense, high-dimensional embeddings and were particularly prone to overfitting and sensitivity to noise.

These results, validated through rigorous cross-validation, confirm that domain-specific embeddings like BioBERT can significantly enhance medical text classification. Future research may further optimize these embeddings or combine them with other transfer learning techniques. Additionally, investigating ensemble methods, advanced neural architectures, or model-agnostic interpretability techniques could yield even greater accuracy and flexibility in real-world healthcare applications.

6 CONCLUSION

This study investigated the efficacy of four supervised machine learning algorithms (Naive Bayes, Decision Trees, Logistic Regression, and SVM) and four text embedding methods (TF-IDF, GloVe, SBERT, and BioBERT) for classifying medical abstracts into predefined categories. The experimental results, validated through a rigorous 5-fold cross-validation framework, highlight several key insights:

- **Impact of Domain-Specific Embeddings:** BioBERT consistently boosted classification performance across most models, underlining the importance of training or fine-tuning embedding models on domain-relevant corpora. In particular, SVM achieved the highest accuracy with BioBERT, demonstrating its robustness and ability to capture nuanced biomedical concepts.
- **Advantage of Contextual Representations:** SBERT showed significant gains over traditional embeddings like TF-IDF

and GloVe, indicating that sentence-level context is critical for accurately interpreting medical terminology and relationships. Even simpler classifiers such as Naive Bayes benefited substantially from SBERT's richer context.

- **Classifier Performance Variability:** SVM and Logistic Regression generally outperformed Naive Bayes and Decision Trees, especially with dense or contextual embeddings. Decision Trees were particularly prone to overfitting in high-dimensional embedding spaces, suggesting limited suitability for complex language representations without further tuning or ensemble methods.
- **Noise Sensitivity and Imbalanced Data:** Models employing domain-specific or contextual embeddings (SBERT, BioBERT) displayed better robustness against noisy text, while TF-IDF and GloVe proved more sensitive to disruptions. Additionally, applying SMOTE and weighted loss functions helped mitigate class imbalance, a common challenge in medical datasets, thereby improving minority-class performance.

7 LIMITATIONS

While our findings are promising, several limitations merit attention:

(1) **Dataset Size and Diversity:** The labeled abstracts used may not fully capture the breadth of clinical text. Labeled data scarcity remains a major hurdle in medical NLP.

(2) **Computational Overheads:** Contextual embeddings such as SBERT and BioBERT demand significant resources, which may limit adoption in smaller healthcare settings.

(3) **Generalizability:** Models trained on abstracts may not seamlessly transfer to other clinical documents (e.g., radiology reports). Fine-tuning or domain adaptation is often necessary.

(4) **Interpretability:** Although methods like Decision Trees and Logistic Regression are relatively transparent, advanced models (e.g., transformer-based SVMs) can act as "black boxes," complicating result interpretation.

8 FUTURE RESEARCH DIRECTIONS

Several promising directions for building on these results include:

(1) **Expanded Datasets and Domains:** Incorporate larger, more diverse text sources (clinical notes, imaging reports) to improve generalization and capture broader medical language.

(2) **Advanced Ensemble Methods:** Combine classifiers or embeddings (e.g., TF-IDF with BioBERT) to improve robustness and handling of domain complexity.

(3) **Neural Architectures and Transfer Learning:** Explore newer transformer-based models (ClinicalBERT, PubMedBERT), multi-task learning, or domain-specific fine-tuning to enhance performance.

(4) **Explainability and Transparency:** Integrate model-agnostic tools (SHAP, LIME) and clinical expert feedback to ensure reliable, interpretable predictions in healthcare settings.

(5) **Real-Time Deployment and Integration:** Optimize models via distillation or specialized hardware to enable real-time use, and integrate with electronic health record systems for practical adoption.

In conclusion, this work highlights the considerable benefits of context-rich, domain-specific embeddings for medical text classification. By addressing computational resources, data constraints, and interpretability, it provides a framework for further innovation in clinical NLP. Collaboration between ML researchers and healthcare professionals will be key to refining automated text analysis and driving better patient care outcomes.

REFERENCES

[1] Ȯzlem Uzuner *et al.*, "Challenges in information extraction from clinical text," *Journal of Biomedical Informatics*, 2008.

[2] D. Demner-Fushman *et al.*, "Automated approaches for clinical question answering," *Journal of the American Medical Informatics Association*, 2009.

[3] P. Yadav and A. Malik, "A study of medical text classification using machine learning," *Healthcare Informatics Research*, 2014.

[4] S. Kim *et al.*, "Comparative analysis of machine learning methods for text classification in healthcare," *Artificial Intelligence in Medicine*, 2017.

[5] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[6] J. Pennington *et al.*, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[7] A. Jagannatha and H. Yu, "Structured prediction models for rnn based sequence labeling in clinical text," in *ACL*, 2016.

[8] B. Chiu *et al.*, "How to train good word embeddings for biomedical nlp," in *ACL*, 2016.

[9] Y. Zhang *et al.*, "Biowordvec: Improved word embeddings for biomedical text," *arXiv:1901.08149*, 2019.

[10] J. Devlin *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[11] J. Lee *et al.*, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, 2020.

[12] E. Alsentzer *et al.*, "Publicly available clinical bert embeddings," in *Proceedings of the 2nd Clinical NLP Workshop*, 2019.

[13] I. Beltagy *et al.*, "Scibert: A pretrained language model for scientific text," in *EMNLP*, 2019.

[14] I. Li *et al.*, "Fine-tuning bert for medical text classification," in *IEEE International Conference on Bioinformatics and Biomedicine*, 2020.

[15] A. Johnson *et al.*, "Mimic-iii, a freely accessible critical care database," *Scientific Data*, 2016.

[16] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48, 1998.

[17] "Naive Bayes classifier - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Naive_Bayes_classifier, [Accessed 28-01-2025].

[18] C. Kim, "Decision Tree Classifier with Scikit-Learn from Python — chyun55555," https://medium.com/@chyun55555/decision-tree-classifier-with-scikit-learn-from-python-e83f38079fea, [Accessed 28-01-2025].

[19] "Support Vector Machine (SVM) Algorithm - Javatpoint — javatpoint.com," https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm, [Accessed 28-01-2025].

[20] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," in *Proceedings of the 13th International Conference on Neural Information Processing*, 2003.

[21] K. Rai, "The math behind Logistic Regression — medium.com," https://medium.com/analytics-vidhya/the-math-behind-logistic-regression-c2f04ca27bca, [Accessed 28-01-2025].

[22] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, 2004, pp. 78–85.

[23] M. M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Systems with Applications*, vol. 106, pp. 36–54, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741741830215X