Bio Medical Text Classification Using ML Models

**Report for MIST 6170 Advance Machine Learning**

**SPRING 2025**

**Submitted by:**

**Abhishek Kumar Singh**

**Nithyashree Bandihalli Rangaswamy**

**Sanjay Sandhosh**

**BUSINESS PROBLEM**

The rapid growth of biomedical literature poses a significant challenge for research teams and clinicians who need to keep up with the latest findings. This project aims to develop an automated classification system that assigns research publications to the appropriate cancer‐type category (Colon Cancer, Lung Cancer, Thyroid Cancer). Such a system can accelerate literature‐review workflows, improve information retrieval, and support decision‐making in research and clinical settings, saving a lot of energy, time and ultimately cost for the companies and for the employees, which can now focus on important things.

**MACHINE LEARNING TECHNIQUES USED**

To address the text classification problem, we applied a progression of methods taught in the course:

* TF-IDF + Classical Models: Term Frequency–Inverse Document Frequency vectorization combined with Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine.
* Ensemble Methods: Random Forest and XGBoost to reduce variance and bias.
* LSTM model
* CNN model

**DATA DESCRIPTION**   
The dataset comprises approximately 6,000 research-paper abstracts from Kaggle, balanced across three cancer types. Each record includes two fields:

* **cancer\_type** (categorical label): Colon, Lung, and Thyroid.
* **research\_paper** (abstract text): No missing values were present, and class proportions were approximately equal.

A screenshot of a computer

AI-generated content may be incorrect.

**DATA PRE-PROCESSING**

* We started with giving the columns meaningful names, checking for null values, cleaning the textual data by removing additional stop words (using nltk library).
* We used test\_train\_split() to split our dataset 3 ways: 60% training set, 20% validation set and 20% testing set.

**EXPLORATORY ANALYSIS**

1. **Distribution of Cancer types:**

**A graph of different colored squares

AI-generated content may be incorrect.**

1. **Average text length by cancer type**

**A graph showing different colored bars

AI-generated content may be incorrect.**

1. **Most frequent words (word clouds) by cancer type**

A close-up of words

AI-generated content may be incorrect.

A close-up of words

AI-generated content may be incorrect.

A close-up of words

AI-generated content may be incorrect.

1. **Top 20 frequent words**

A graph of different colored bars

AI-generated content may be incorrect.

**MACHINE LEARNING PROCESS**

The workflow consisted of:

1. **Data Cleaning:** Lowercasing, removal of punctuation and digits, and elimination of English stop words (plus domain-specific terms “et”, “al”, “aal”).
2. **Feature Engineering:** Tokenization via CountVectorizer and TF-IDF weighting with TfidfTransformer.
3. **Data Splitting**: Stratified three-way split into training (60 %), validation (20 %), and test (20 %) sets.

A black numbers on a white background

AI-generated content may be incorrect.

1. **Baseline Modeling:** Fitting and evaluating Multinomial Naive Bayes, Logistic Regression, and SVM on the TF-IDF features.
2. **Ensemble Methods:** Training Random Forest and XGBoost on the same features.
3. **Transformer Fine-Tuning:** Loading a pretrained BERT model, tokenizing abstracts, and fine-tuning for three epochs.(Only able to run epoch with 95% error before getting compatibility issues.)

A diagram of a process

AI-generated content may be incorrect.

**INTERPRETATIONS AND RESULTS**

1. **Naive Bayes:** Validation accuracy ~ 91 %; class-level F1 ranged 0.87–0.99.
2. **Logistic Regression:** Validation accuracy ~ 93 %; improved recall on Colon\_Cancer.
3. **SVM:** Validation accuracy ~ 92 %; nearly identical to Logistic Regression.
4. **Random Forest & XGBoost:** Marginal gains over Logistic Regression; XGBoost required label encoding but achieved ~ 100 %.
5. **LSTM:** With an accuracy of 85%, this model isn’t expressive enough and might need some parameter changes to get more accurate results.
6. **CNN:** 97% accuracy is very high & likely means the model is overfitting pr it could just mean that the model is very good at picking up local patterns in text, via sliding convolutional filters.

|  |  |  |
| --- | --- | --- |
| **Model** | **Validation Accuracy** | **Note** |
| **Multinomial Naive Bayes** | 91% | Strong baseline |
| **Logistic Regression** | 93% | Highest among linear classifiers |
| **Support Vector Classifier (SVC)** | 92% | Nearly identical to Logistic Regression |
| **Random Forest** | 100% | **Likely overfitting**—unpruned trees |
| **XGBoost** | 100% | Likely overfitting |
| **LSTM** | 85% | 1 LSTM layer |
| **CNN** | 97.29% |  |

**MANAGERIAL AND PRACTICAL IMPLICATIONS**  
An accurate, automated classifier can reduce literature‐review workload by directing researchers and clinicians to the most relevant publications. Embedding this system in digital libraries, research dashboards, or clinical-decision platforms would streamline workflows, improve resource allocation, and support evidence-based practice. Future work could extend to additional cancer types, adapt to real-time document streams, or integrate more specialized biomedical embeddings.

**CHALLENGES ENCOUNTERED**

1. **Inflated Accuracy on Structured Data**  
   Our dataset is well structured and well-formatted, which leads to deceptively high accuracy scores that may not hold up on real-world, noisy inputs.
2. **Heavy Resource Requirements**  
   Fine-tuning transformer models (e.g., BERT) proved extremely time-consuming and memory-intensive on our hardware, resulting in repeated RAM and GPU out-of-memory errors.
3. **Lack of Reproducibility Across Environments**  
   In a separate, we reached 97 % accuracy with BERT, but that environment couldn’t be replicated in our core workflow. Consequently, we decided to remove BERT results from the final report to maintain consistency.

**CONCLUSION**

* The Random Forest’s & XGBoost’s 100 % score on validation almost certainly reflects overfitting (each tree memorizing its bootstrap sample); consider limiting tree depth or using cross-validation to mitigate this.
* Since the CNN model achieved the highest validation accuracy, we selected it for evaluation on the final test dataset and found that the test accuracy is 97.16%.