Cancer Research Abstracts: Topic Classification Using Machine Learning Algorithms

Final Report

John Zumel and Samantha Zygmont

IST 736: Text Mining

March 20, 2025

1. Introduction

Cancer is one of most prominent health concerns worldwide, with instances rising considerably over the past few decades. While incidence of cancer cases have grown, mortality rates have remained fairly stable or even declined for certain cancers (Devesa). This data suggests that improvements in cancer detection and treatments from research and clinical practice is having real effects on cancer treatment. Therefore, cancer research remains a critical focus for clinicians and researchers for continuing to improve patient outcomes. As cancer is increasingly becoming a leading cause of premature death in many countries, even surpassing heart disease in certain regions, the burden of cancer on global health systems is expected to grow (Bray). Hence, the need for comprehensive and improved cancer research remains a priority worldwide.

The starting point of any research study involves consulting literature to understand the disease and what systems have been effective in treatment. Cancer treatments are different from many other diseases due to the unique nature of each type of cancer. With the shift in cancer research towards molecularly targeted treatments, the focus has shifted towards understanding the genetics and biology of both the patient and their specific cancer to ensure more effective treatments (de Bono). Therefore, in this paper we sought to create a model capable of helping researchers filter literature specific to their field of work. The goal of the project is to use a series of abstracts from colon, lung, and thyroid cancer papers to train a model to quickly classify research papers based on the cancer type. Abstracts were used as the basis for sorting these papers because abstracts summarize the key findings of research papers. This model will help researchers, healthcare professionals, and students streamline the process of discovering relevant research papers and ultimately improving cancer patient outcomes.

2. About the Data

2.1 Model Training Data

This data set was sourced from Kaggle and entitled "Medical Text Dataset - Cancer Doc Classification" (FALGUNIPATEL19). It consists of abstracts from scientific research papers pertaining to colon cancer, lung cancer, and thyroid cancer. The criteria for selecting these papers were that they were on the longer side, with 6 or more pages in total. The idea behind selecting longer papers was that the papers were more likely to contain comprehensive information summarized in the abstract. The data set contains 7,569 entries with two columns: "Research Paper Text," which includes the abstracts, and "Class Labels," which labels the text as either pertaining to "Colon Cancer," "Lung Cancer," or "Thyroid Cancer." The large size of this data set and well labeled abstracts made this a very suitable dataset to train our model.

2.2 Model Test Data

This data set was also sourced from Kaggle and was titled "Cancer Papers Dataset" (POURIA1206). This dataset includes a column called "Title" which is the title of the paper, "Abstract" which is the abstract of the research paper, and "Label" which labels the data as either "Colon Cancer," "Lung Cancer," or "Thyroid Cancer." There are a total of 900 abstracts in this data set. This data set is ideal for use as the test data set because it has a large sample size and abstracts labeled by the cancer type so we can determine the accuracy of our model performance.

3. Methods & Models

3.1 Loading Data and Initial Exploratory Data Analysis

For this analysis, Python was used through Google Colab. The entire data set was initially loaded into Google Colab from the .csv file entitled "alldata_1_for_kaggle.csv," as a pandas data frame. Once the data was loaded in, the columns were properly labeled as "Class" and "Abstract,"

and the “Serial Number” column was dropped, as it was not necessary for our analysis. Once the data was loaded in, exploratory data analysis was performed on the data to get a better understanding of what the data looked like. First, a distribution plot was created using seaborn and matplotlib to visualize the distribution of the different cancer types in the data set. Next, there was a histogram created to visualize the distribution of abstract lengths based on word count for each of the different cancer types. Lastly, there was a boxplot created to visualize the length of each abstract based on the cancer type.

3.2 Data Pre-Processing and Vectorization

Once the data was loaded and explored, the abstracts needed to be pre-processed. There was a pre-processing function created in which the text was converted to lowercase, the special characters were removed, the text was tokenized, and the common “english” stopwords were removed. Then, this preprocessing pipeline was applied to the abstract column of the data frame. The data frame was then split into X and y variables, with X being the pre-processed abstracts and y being the classes. The X data was then vectorized using TF-IDF vectorization, and finally the data was split into training and test sets. After the data was pre-processed and vectorized, there were word clouds created for the top words based on each cancer type.

3.3 Baseline Models

Once the data was pre-processed and vectorized, it was tested on a series of different baseline models. The first model tested was a logistic regression model, in which the parameters were set to C=0.1, solver=saga, penalty=l2, and max_iter=1000. Then the model was fit to our training data and the training and validation accuracies were calculated and put into a pandas dataframe called “model_accuracies” to keep track of all the accuracies. Next, another logistic regression model was tested, except the C was increased to 1 and the other parameters were left

the same to see how increasing the C value affected the model's accuracy. The training and validation accuracies were then calculated for this new model and added to the "model_accuracies" data frame. The next model tested was a Multinomial Naive Bayes (MNB) Model with alpha=0.1 and fit_prior=True. The model was then trained with the data and the training and validation accuracies were again calculated and added to the "model_accuracies" data frame. Then another MNB model was trained based on the data with the alpha increased to 1 and fit_prior staying the same. The model was then trained again to see how the increase in alpha affected the accuracies of the model. Once all of the training and validation accuracies were calculated for the Baseline Models, a bar plot was created summarizing the training and validation accuracies of each of the initial models tested.

3.4 Intermediate Models

After initially testing the baseline models, we moved on to mode intermediate models. The first type of models we tested were SVM models. The first SVM models had the parameters C=0.1, kernel=linear, and class_weight=balanced. Then the model was trained with the abstract data and the training and validation accuracies were calculated and put into a data frame called "inter_model_accuracies." For the next SVM model, the linear kernel was used again but the C was increased to 1. The training and validation accuracies were again calculated and added to the "inter_model_accuracies" data frame. There were then 4 more SVM models created and tested: 2 using the rbf kernel and 2 using the poly kernel with C=0.1 in one model and C=1 in the other model for each kernel. This was done to determine the optimal kernel and C parameters for the SVM model using our data. Training and validation scores were again calculated for each of these SVM models and added to the "inter_model_accuracies" data frame. Finally, there was a Random Forest Model created with the parameters: n_estimators=100, max_depth=10,

min_samples_split=10, min_samples_leaf=5, random_state=42, and n_jobs=-1. The training and validation accuracies were then also calculated and added to the "inter_model_accuracies" data frame. Once all of the training and validation accuracies were calculated for the intermediate models, a bar plot was created summarizing the training and validation accuracies of each of the initial models tested.

For the Random Forest Models, there were some additional calculations performed to determine if there was any overfitting occurring. To do so, a 10-fold cross validation score was calculated for the random forest model. The cross validation accuracy score, the average cross-validation accuracy score, and the standard deviation of the cross validation accuracy were then calculated and stored in a data frame called "cv_results." Then another Random Forest Model was created but with constraints to prevent overfitting. These constraints were set up using Stratified K-Folds which ensure that the class distribution is maintained in each fold. Then, the cross validation accuracy score, the average cross-validation accuracy score, and the standard deviation of the cross validation accuracy were again calculated for the Random Forest Model with constraints and stored in a data frame called "cv_results."

3.5 Testing Models

Once we had the initial models, we wanted to run our test data through the models to see how well they performed. First, the test data in file "data2.xlsx" was loaded in as a pandas data frame. The columns in the test data set were then renamed so they were consistent with that of the training data. The new data was then pre-processed and vectorized using the same methods as the data used to train the model.

Once the data was pre-processed it was run through the four baseline models. For each of the models, the test data was run through the model and used to create an additional column called

"Predicted Class." The predicted classes were then compared to the actual classes to determine the test accuracies of each model. This information was added to the model_accuracies data table. This data was then plotted on a bar plot with each model and its training accuracy, validation accuracy, and test accuracy. These exact steps were repeated for the intermediate models, stored in the inter_model accuracies, and plotted.

3.6 Fine-Tuning Models

Once the models performance was tested, the top performing models hyperparameters were tuned to see if model performance would improve. Based on the initial findings, the Logistic Regression models had the highest test accuracies, so they were explored using different hyperparameters. We decided to vary the C value, seeing if a C of 0.1, 0.2, or 0.5 was optimal for model performance. We also varied the solver, trying "saga," "liblinear," and "elasticnet" to see how those impact model performance. Lastly we varied the L1 ratio to see if that had any impact on model performance. Once these hypertuned models were made, the test data was run and the training accuracy, validation accuracy, and testing accuracy was calculated, stored in a data frame, and plotted. Finally, there were confusion matrices created for each of the fine-tuned models to determine how well the model's performed.

4. Experimental Design & Investigation Strategy

The ultimate goal of this analysis was to create a robust model capable of classifying cancer types based on scientific paper abstracts. In order to achieve this goal, we wanted to try as many models as possible to determine which models perform best with our data. The models we decided to train with our data were Logistic Regression, Multinomial Naive Bayes, SVM, and Random Forest models. We selected these models due to their effectiveness in text classification tasks. For our initial pre-processing, we decided to keep our steps very basic but effective, as to not over-

process the data. We then chose to use TF-IDF vectorization because it scores the most important words in the abstracts which is important for our models to capture the most important words for classification.

For our models, we decided to start with what we called "Baseline Models" which consisted of two logistic regression models and two MNB models. We then moved onto more "Intermediate Models" which were SVM models with different parameters and random forest models. We used the training accuracy and validation accuracy as a measure of comparison between each of the models. Once we compared each of these models, we used a test data set to test each model's performance. We determined the test accuracy based on the predicted classes and actual classes. Finally, once we determined which model's performed the best with the test data, we tuned the hyperparameters to further improve our top performing model's accuracy. Table 1 and Table 2 is provided in the Appendix section, highlighting the baseline and intermediate model hyperparameters and updates made during fine-tuning.

5. Results

The initial exploratory data analysis results are illustrated in the figures below. In Figure 1, the distribution plot of cancer types in the dataset are shown. Based on the plot, it is clear that the most number of abstracts were of Thyroid Cancer with about 2,800 documents, then Colon Cancer with about 2,500 documents, and finally Lung Cancer with about 2,250 documents. In Figure 2, the distribution of abstract lengths for each cancer type is illustrated. In this plot, the Lung Cancer abstracts seem to have the smallest length with the majority having about 1,000 words in length. The Colon Cancer and Thyroid Cancer abstracts seem to have very similar lengths with their bars overlapping, with the majority of documents seeming to have about 4,500 words for both, with Thyroid Cancer seeming to be slightly smaller in length than Colon Cancer. Lastly,

Figure 3 shows the box plot of the abstract lengths versus cancer type. This plot confirms what we saw in Figure 2, with Lung Cancer abstracts having the majority of documents with smallest length but the greatest range. Colon Cancer seems to have the largest overall abstract length, with its average just slightly above that of Thyroid Cancer, which we observed in Figure 2 as well.

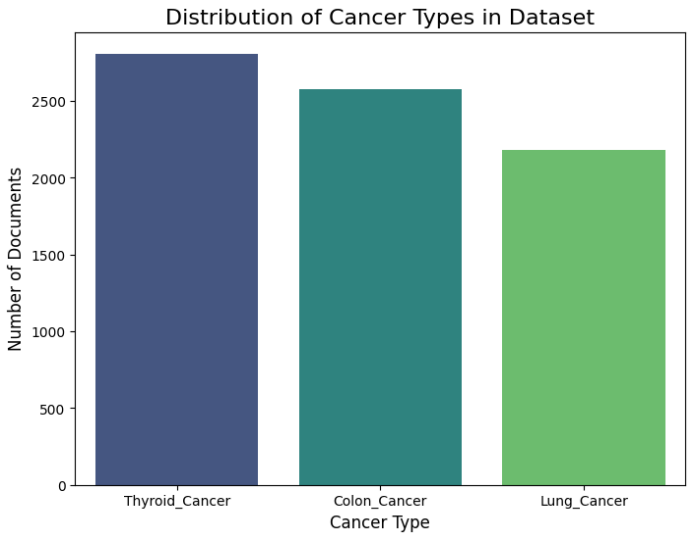

Figure 1: Distribution Plot of Number of Documents for Each Cancer Type

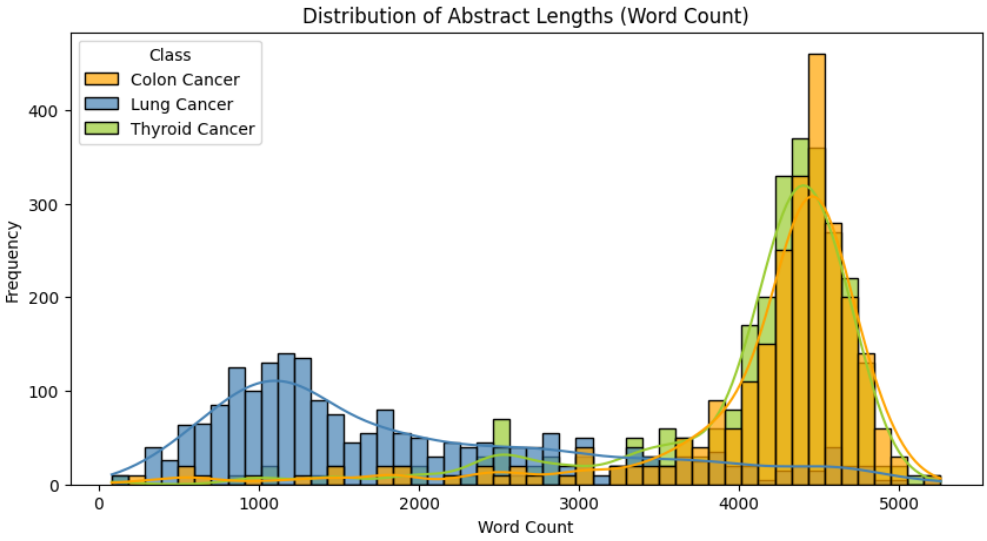

Figure 2: Distribution of the Abstract Length for Each of the Cancer Types

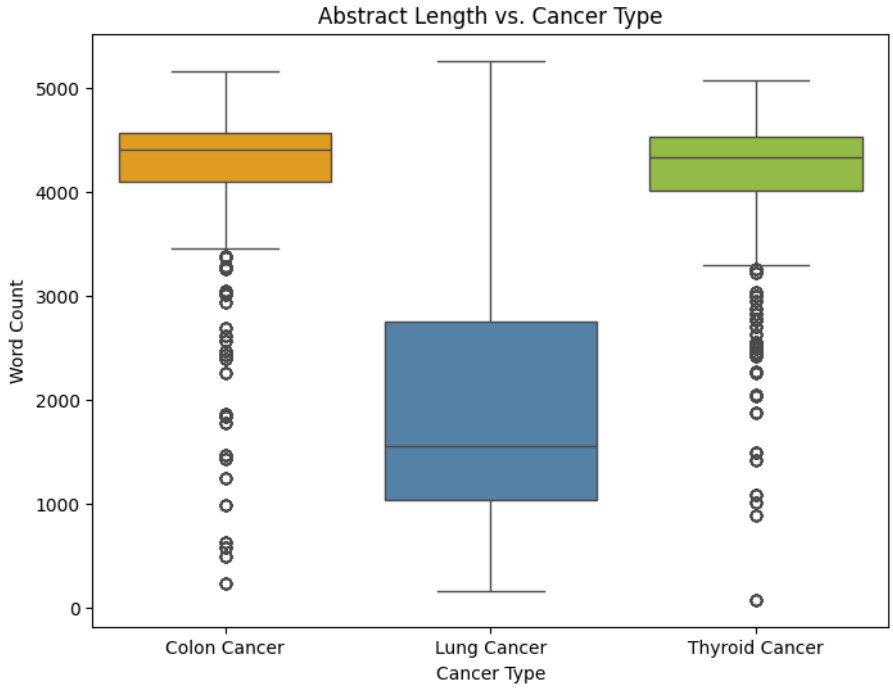

Figure 3: Box Plot Comparison of Abstract Lengths for Each Cancer Type

The following figures illustrate the word clouds constructed to show the top words in the abstracts of each cancer type, after pre-processing and vectorization. Figure 4 shows the word cloud for the Colon Cancer abstracts with some of the top words being: “cancer,” “cell,” “breast,” “creative,” and “colorectal.” Next, Figure 5 shows the word cloud for the Thyroid Cancer abstracts with some of the top words being: “cancer,” “cell,” “breast,” “lung,” and “expression.” Finally, Figure 6 shows the word cloud for Lung Cancer with some of the top words being: “lung,” “breast,” “cancer, “common,” and “cell.” The most common words across all of the word clouds are to be expected with many words pertaining to cancer such as “cell,” “cancer,” and “expression.” There were also the words “breast” and “lung” in all of the word clouds, implying that other cancer types were also discussed in these abstracts.

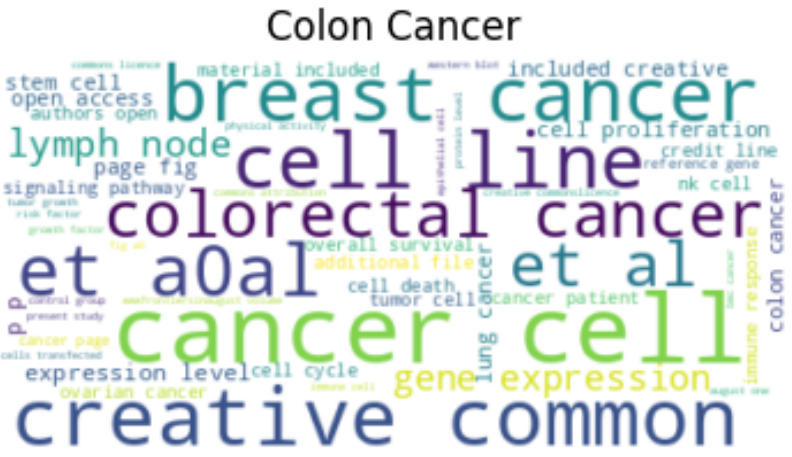

Figure 4: Word Cloud of Most Common Words in Colon Cancer Abstracts

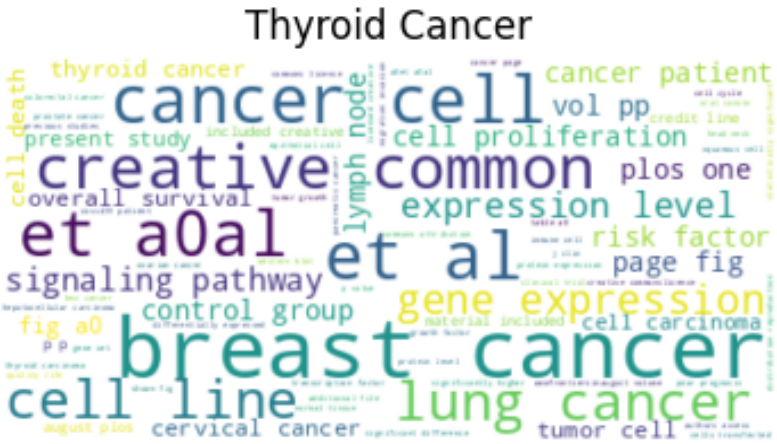

Figure 5: Word Cloud of Most Common Words in Thyroid Cancer Abstracts

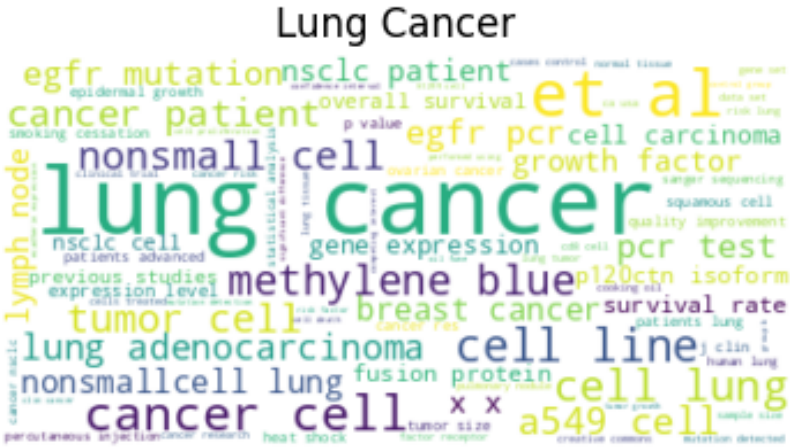

Figure 6: Word Cloud of Most Common Words in Lung Cancer Abstracts

The next figures illustrate the results of the model testing. In Figure 7, the training and validation accuracies are shown for the baseline models run. The model with the highest accuracies was the Logistic Regression Model with C=1, with a training accuracy of ~96% and validation accuracy of ~94%. The Logistic Regression Model with C=0.1 performed the next best with a training accuracy of ~93% and a validation accuracy of ~91%. The next best model in terms of accuracy was the MNB model with an alpha=0.1 which had a training accuracy of ~92% and a validation accuracy of ~91%. Lastly, the model that had the lowest accuracies was the MNB model with the alpha=1, which had a training accuracy of ~91% and a validation accuracy of ~89%. Therefore from the initial modeling, each of these models had very high accuracy, >89%, but the logistic regression models outperformed the MNB models with our data set. Table 3 stores numerical values of baseline model accuracies for comparison and can be found in the Appendix section.

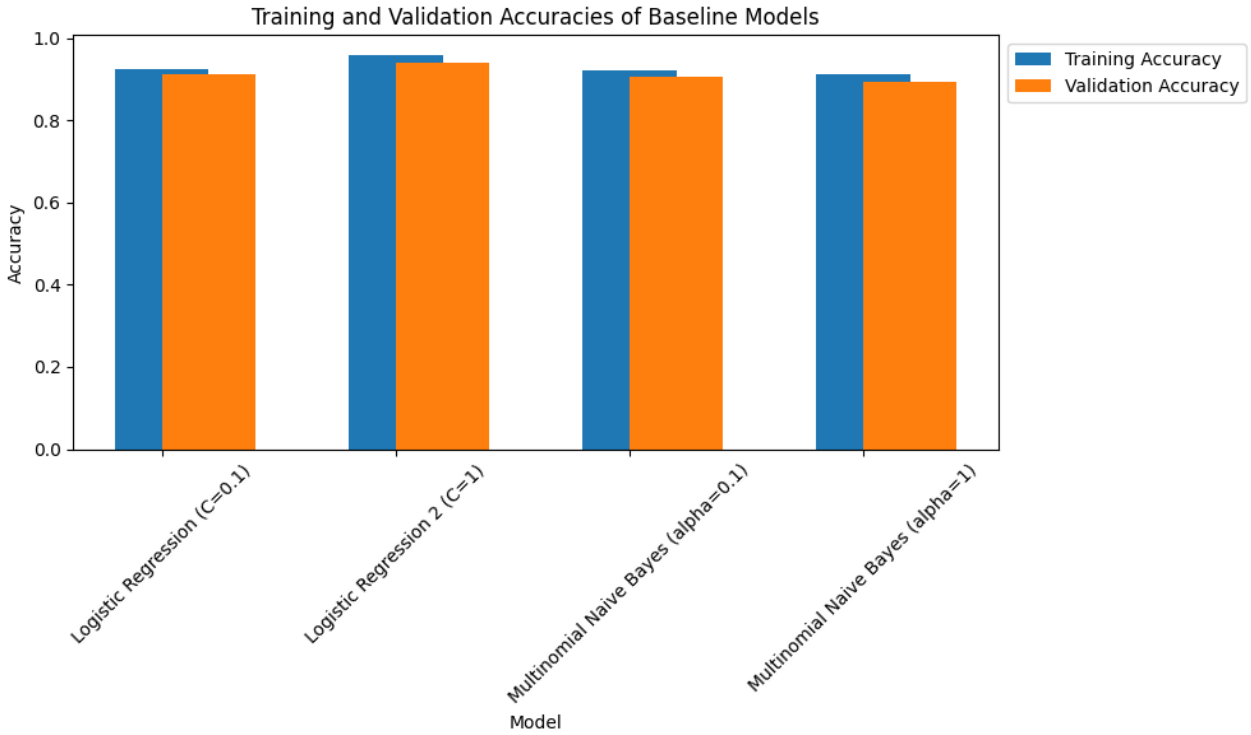

Figure 7: Training and Validation Accuracies for the Baseline Models

The next figure shows some of the results of testing more intermediate models. Figure 8 illustrates the comparison of the training and validation accuracies for the intermediate models. The model which performed the best was the Random Forest models with a training accuracy of ~99.8% and validation accuracy of ~99.5% for the flexible model and a training accuracy of ~98.8% and validation accuracy of ~98.7% for the conservative model. The SVM model did not perform as well as the Random Forest model, but it still had very high accuracies. The optimal parameters for the SVM model were C=1 and kernel=rbf, with a training accuracy of ~95% and validation accuracy of ~92%. The SVM model with the linear kernel and C=0.1 performed the worst of all the SVM models with a training accuracy of ~92% and validation accuracy of ~91%. Each of these intermediate models performed very well with our data, with accuracies >90%, but the model which performed the best was the Random Forest model. The cross-validation scores were then calculated for the Random Forest model with and without constraints for overfitting. The value of the average cross-validation remained the same for the Random Forest model with and without constraints for overfitting, yielding an average cross validation score of 99.3%. Table 3 stores numerical values of intermediate model accuracies for comparison and can be found in the Appendix section.

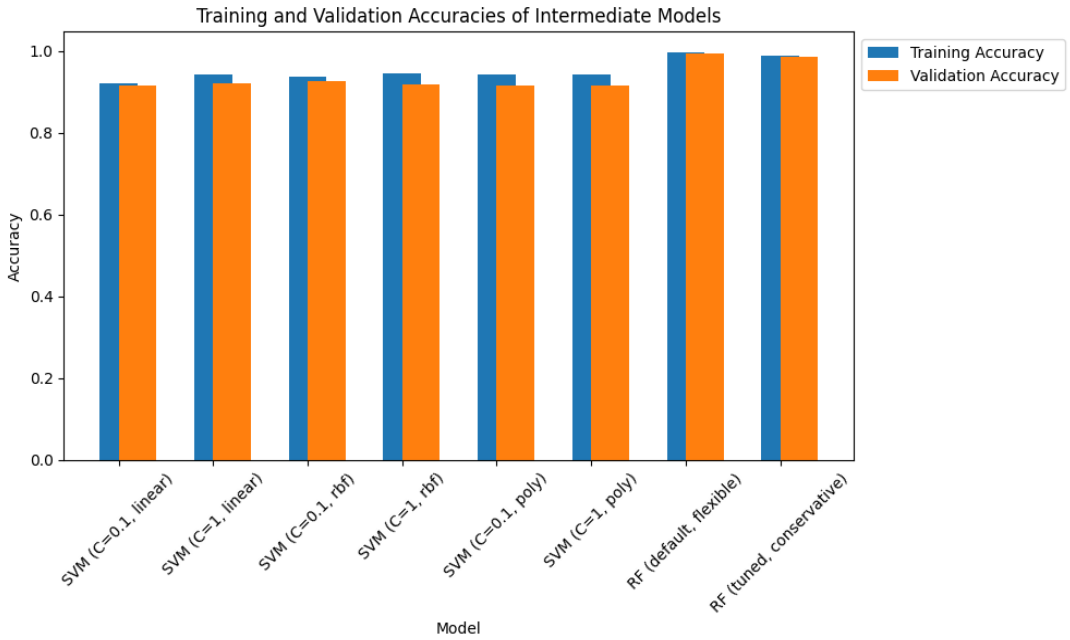

Figure 8: Training and Validation Accuracies for the Intermediate Models

The following figures illustrate the test accuracies of the model's performance with the test data. Illustrated in Figure 9, of the baseline models tested, the logistic regression (LR) model's again outperformed the MNB models in terms of test accuracy. The LR model with a C=0.1 had the best test accuracy of 71.6% and the MNB model with the alpha=0.1 had the worst test accuracy of 55.7%. For the intermediate models tested, there were some interesting findings. As shown in Figure 10, the Random Forest models initially showed the highest training and validation accuracies, however after testing the models, their test accuracy of 33% was extremely low. The SVM model with C=0.1 and the linear kernel performed the best of all the intermediate models, with a test accuracy of 68.6%. The initial model testing showed that the logistic regression models had the best performance. Table 4 stores numerical values of model accuracies for comparison and can be found in the Appendix section.

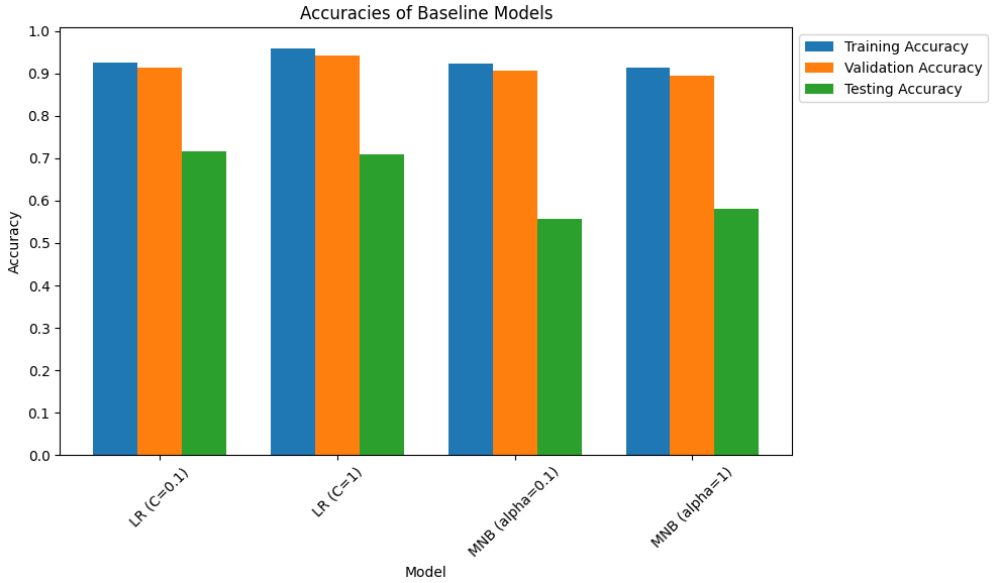

Figure 9: Training, Validation, and Test Accuracies for the Baseline Models

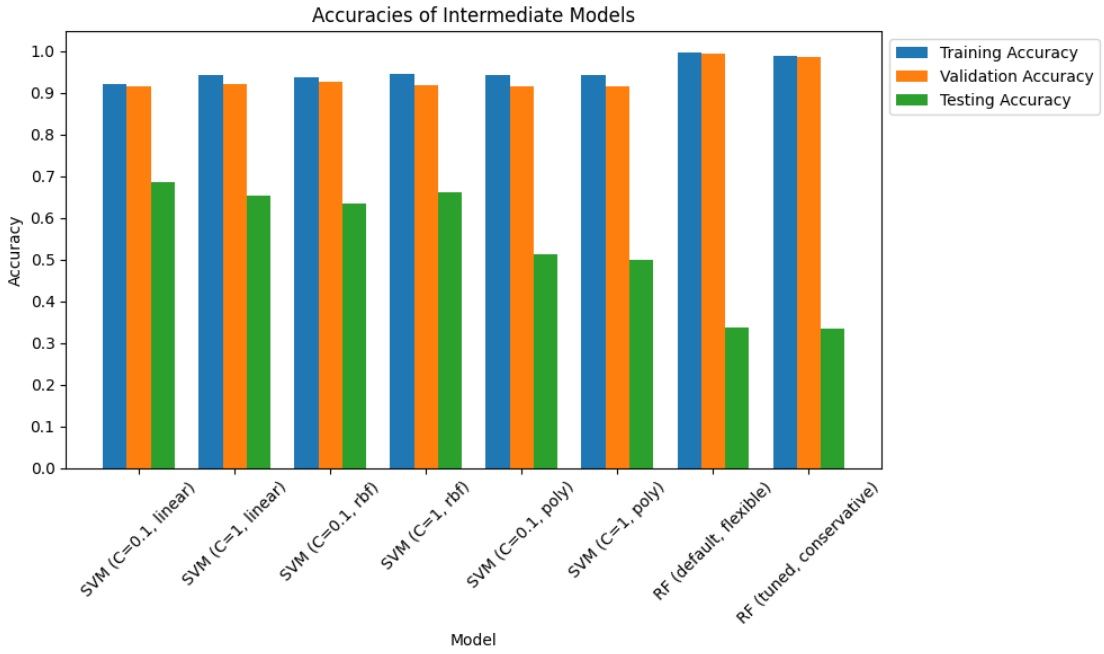

Figure 10: Training, Validation, and Test Accuracies for the Baseline Models

Based on the strong performance of the Logistic Regression models, the parameters were tuned to increase accuracy. Based on the results in Figure 11, each of the tuned models had test accuracies >79% which was very promising. The model with the best test accuracy was the tuned LR model with C=0.1, solver=liblin, and penalty=l1, with a test accuracy of 85.7%. Though the

most balanced accuracies were found in the last three tuned LR models: (C=0.1, elasticnet, l1_ratio=0.4), (C=0.1, elasticnet, l1_ratio=0.4, class_weights=balanced), and (C=0.1, liblinear, penalty=l1, class_weights=balanced). These three models were able to generalize particularly well with new unseen data and shared very similar accuracies across the training, validation, and testing phases. Therefore, these three models might offer a *safer* alternative to Tuned LR Model 2 (C=0.1, solver=liblin, and penalty=l1).

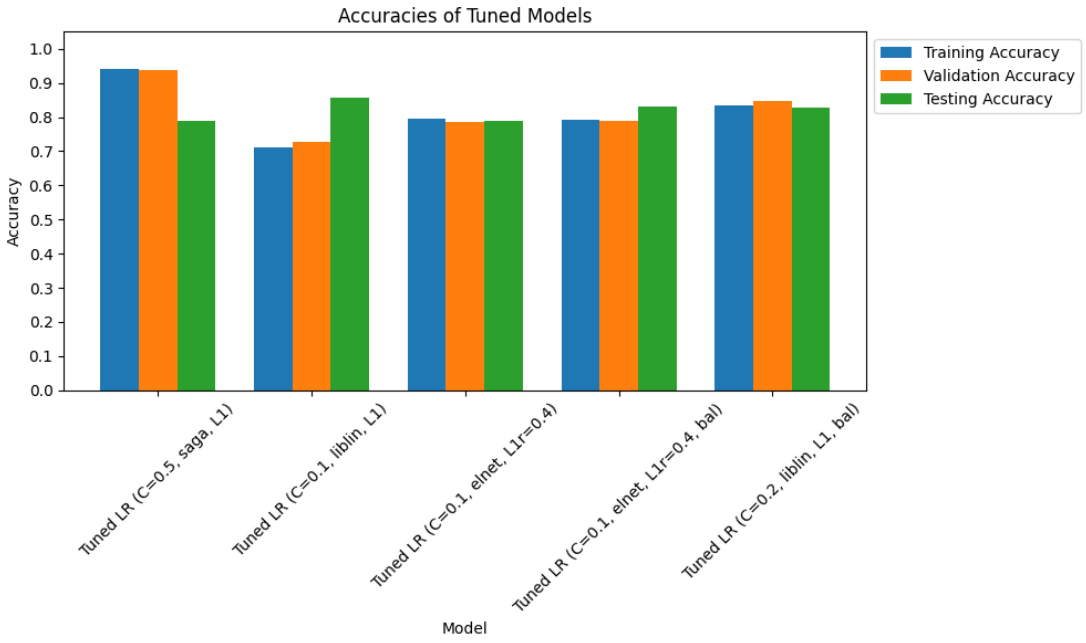

Figure 11: Training, Validation, and Test Accuracies for the Tuned Models

Finally, the tuned models were further tested with confusion matrices, which illustrated how well the models predicted each category. As seen in figures 12-16, each of the model's had the highest accuracy with predicting either Thyroid or Colon Cancer, and the lowest accuracy with predicting Lung Cancer. Figure 13, illustrates the model with the highest test accuracy. This model had the highest recall and f1-score for predicting Lung Cancer of all the tuned models. This model correctly predicted the classification of 771 abstracts out of a total of 900 in the test data. This model shows great promise for initial steps in its implemented use.

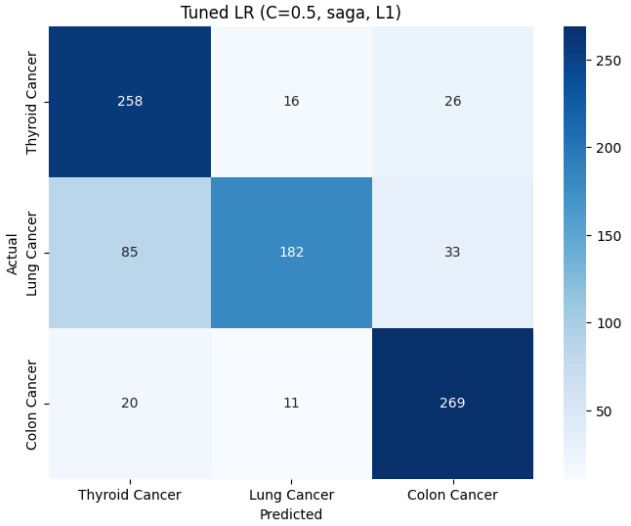

Figure 12: Confusion Matrix for Tuned Model 1

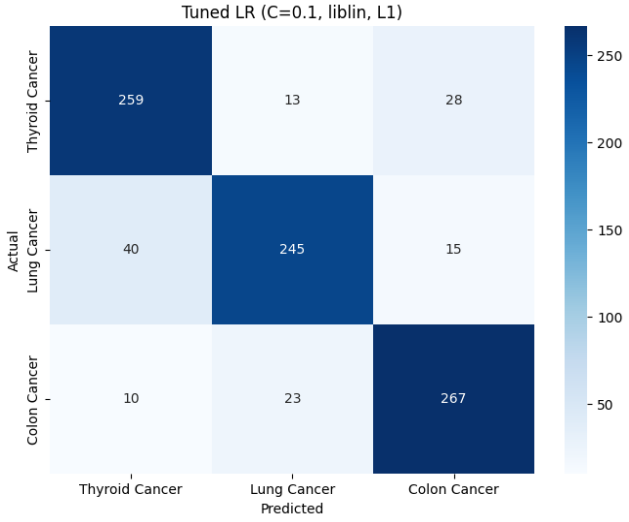

Figure 13: Confusion Matrix for Tuned Model 2

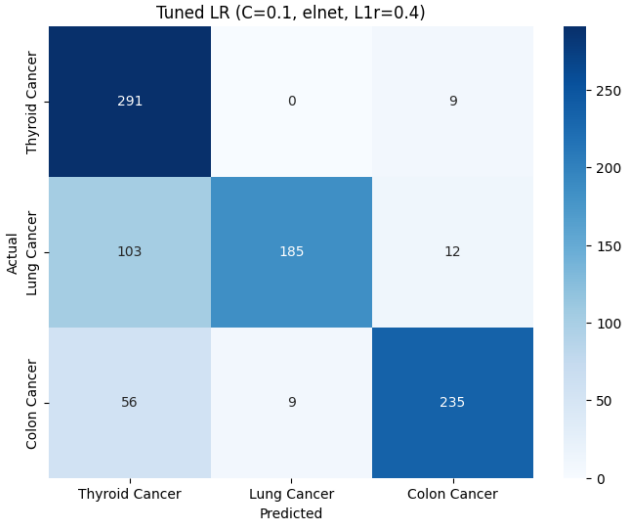

Figure 14: Confusion Matrix for Tuned Model 3

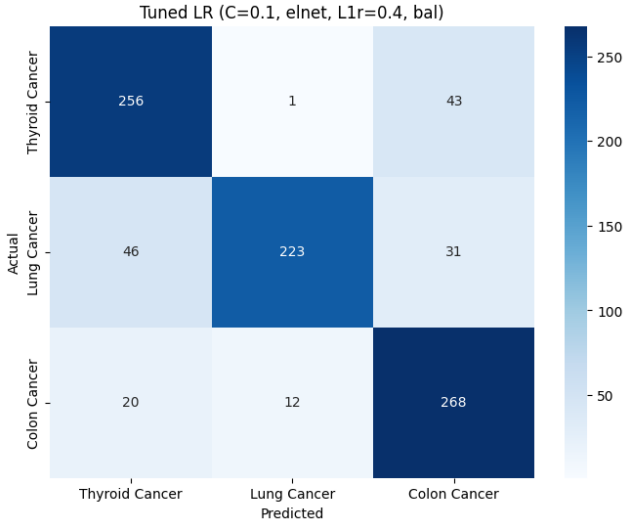

Figure 15: Confusion Matrix for Tuned Model 4

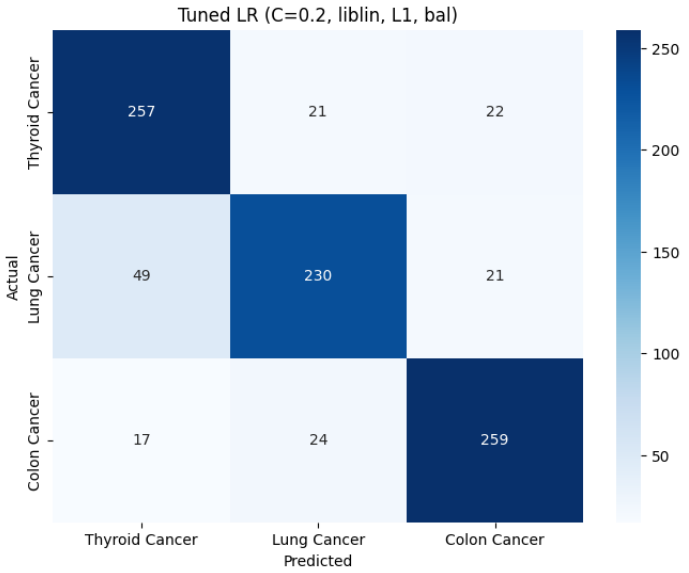

Figure 16: Confusion Matrix for Tuned Model 5

6. Conclusion

Overall, the results of the analysis show immense promise for the first steps towards creating a model capable of classifying scientific abstracts based on cancer type. The tuned logistic regression models yielded very promising results. Model 2, from the tuned models, yielded a very high test accuracy of 85.7%. However, a *safer* alternative would be tuned Model 5. This model generalizes well and boasts very balanced accuracies among all stages. The confusion matrices further supported this data by showing how well the tuned models performed across all the cancer types. Interestingly, all the tuned models performed the worst when classifying Lung Cancer. As we saw in the earlier word clouds in the initial EDA, one possible explanation could be that the word "lung" was discussed in the abstracts across all cancer types. It is possible the prominence in the word "lung" and other words across all the cancer types made it more challenging for the model to differentiate. Another possible reason for the lower performance for classifying lung cancer abstracts is that we noted in our EDA that there was the smallest number of abstracts pertaining to Lung Cancer and they were also the shortest in length in the training dataset. In future studies we

would like to make sure there is an even distribution of all cancer types used when training the model.

There were some limitations with this study that made it challenging for us to perform these analyses. The first limitation was the large size of the data. While it was important to train our model with a large data set, there were extremely long run times for some of our models. Specifically our SVM models had run times of 5-10 minutes per model, making the process of model selection long and tedious. Another limitation of our study was model overfitting. Specifically when we ran the Random Forest models, we were getting training and validation accuracies of nearly 100%, however when we tested the models, the test accuracy was only about 33%.

In future studies we would like to enhance our model by expanding the diversity of our data set. We would like to source abstracts from other cancer types such as breast cancer, prostate cancer, stomach cancer, and more to enhance the robustness of our model. Additionally we would like to incorporate some clinical data in our model to further differentiate the model's ability to classify the abstracts by cancer type. We also plan to explore more advanced features such as incorporating deep learning models to help tune the model to pick up on any nuances which may be present in our data. Most importantly, we would like to collaborate with scientific researchers, clinicians, students, and more to understand their needs and make this model meaningful and practical in a real world setting.

7. Appendix

1. Devesa, Susan S., et al. "Recent Cancer Trends in the United States." *JNCI: Journal of the National Cancer Institute*, vol. 87, no. 3, Feb. 1995, pp. 175–182, https://doi.org/10.1093/jnci/87.3.175.

2. Bray, Freddie, et al. "The Ever-Increasing Importance of Cancer as a Leading Cause of Premature Death Worldwide." *Cancer*, vol. 127, no. 16, 2021, pp. 3029-3030, https://doi.org/10.1002/cncr.33587.
3. de Bono, J. S., and Alan Ashworth. "Translating Cancer Research into Targeted Therapeutics." *Nature*, vol. 467, 30 Sept. 2010, pp. 543–549.
4. FALGUNIPATEL19. *Medical Text Dataset - Cancer Doc Classification*. Kaggle, 2022, https://www.kaggle.com/datasets/falgunipatel19/biomedical-text-publication-classification?resource=download
5. POURIA1206. *Cancer Papers Dataset*. Kaggle, 2025, https://www.kaggle.com/datasets/pouria1206/cancer-papers-dataset?resource=download
6. Experimental Design Tables

Model	**Description**
Baseline	Logistic Regression and Multinomial Naive Bayes
Model 1	LR: C=0.1, max_iter=1000
Model 2	LR: C=1, max_iter=1000
Model 3	MNB: a=0.1
Model 4	MNB: a=1
Intermediate	Support Vector Machines and Random Forest
Model 5	SVM: kernel=linear, C=0.1
Model 6	SVM: kernel=linear, C=1
Model 7	SVM: kernel=rbf, C=0.1
Model 8	SVM: kernel=rbf, C=1
Model 9	SVM: kernel=poly, C=0.1
Model 10	SVM: kernel=poly, C=1

Model 11	RF: n_estimators=150, max_depth=10, min_samples_split=15, min_samples_leaf=10
Model 12	RF: n_estimators=300, max_depth=8, min_samples_split=20, min_samples_leaf=15, class_weight="balanced"

Table 1: Experimental Design

Model	Description	Change
Fine-Tuned	Logistic Regression	
FT Model 1	Tuned LR: (C=0.5, saga, L1)	Update Base Model 1, increase C, L2->L1, no iteration cap
FT Model 2	Tuned LR (C=0.1, liblinear, L1)	Update Base Model 1, saga->liblinear, L2->L1, no iteration cap
FT Model 3	Tuned LR (C=0.1, elasticnet, L1_ratio=0.4)	Update Base Model 1, L2->elasticnet with L1_ratio = 0.4
FT Model 4	Tuned LR (C=0.1, elasticnet, L1_ratio=0.4, bal)	Update FT Model 3, balanced weight class
FT Model 5	Tuned LR (C=0.2, liblinear, L1, bal)	Update FT Model 2, increase C, balanced weight class

Table 2: Fine -Tune Model Design

7. Numeric Model Accuracies Tables

Model	Training Accuracy	Validation Accuracy	Testing Accuracy
Baseline			
Model 1	92.59%	91.35%	71.44%
Model 2	96.00%	94.19%	71.00%
Model 3	92.35%	90.62%	55.67%
Model 4	91.33%	89.43%	58.11%
Intermediate			
Model 5	92.21%	91.55%	68.56%
Model 6	94.34%	92.27%	65.44%

Model 7	93.64%	92.73%	63.44%
Model 8	94.60%	91.88%	66.22%
Model 9	94.25%	91.61%	51.22%
Model 10	94.35%	91.55%	49.89%
Model 11	99.79%	99.54%	33.67%
Model 12	98.84%	98.68%	33.33%

Table 3: Baseline and Intermediate Model Accuracies

Model	Training Accuracy	Validation Accuracy	Testing Accuracy
Fine-Tuned LR			
FT Model 1	94.20%	93.79%	78.67%
FT Model 2	70.97%	72.79%	**85.67%**
FT Model 3	79.56%	79.06%	79.00%
FT Model 4	**79.36%**	**78.86%**	**83.00%**
FT Model 5	**83.44%**	**84.68%**	**82.89%**

Table 4: Fine-Tuned Model Accuracies