NLP Cancer Classification

Final Report on : *Machine Learning Analysis of The Top 10 Most Frequent Cancer Types in Men, Women, and Mixed Populations*

By Group A (Anand Enkhbayar , Amina Enkhbayar and Yu Chun Lin)

Introduction

Cancer is a complex and devastating disease that affects millions of people worldwide. Understanding the patterns, causes, and potential treatments for different types of cancer is crucial for improving patient outcomes. In this report, we present the findings of our machine learning analysis on the top 10 most frequent cancer types in men, women, and both genders combined. We utilized a dataset of 300,000 scientific articles downloaded from PubMed, with 100,000 articles for each cancer category.

Methodology

Our analysis involved a combination of natural language processing (NLP) techniques and machine learning algorithms. Our datasets are based on the World Cancer Research Fun International 2020 data on men's, women's, and mixed datasets. The pipeline is based on the notebooks provided by the professor. We made some alterations to the original code to accommodate for outside testing sections.

Data Collection

We downloaded the scientific articles in text format from PubMed, focusing on the top 10 most frequent cancer types in men, women, and mixed populations. Each cancer category consisted of 100,000 articles, resulting in a total dataset of 300,000 articles.

Top 10 cancers in mixed populations:

Rank	*Cancer*	*New cases in 2020*	*% of all cancers*
All	cancers*	18,094,716	--
1	Breast	2,261,419	12.5
2	Lung	2,206,771	12.2
3	Colorectal**	1,931,590	10.7
4	Prostate	1,414,259	7.8
5	Stomach	1,089,103	6.0
6	Liver	905,677	5.0
7	Cervix uteri	604,127	3.3
8	Esophagus	604,100	3.3
9	Thyroid	586,202	3.2
10	Bladder	573,278	3.2

Top 10 cancers in men:

Rank	*Cancer*	*New cases in 2020*	*% of all cancers*
All	cancers*	9,342,957	--
1	Lung	1,435,943	15.4
2	Prostate	1,414,259	15.1
3	Colorectal**	1,065,960	11.4
4	Stomach	719,523	7.7
5	Liver	632,320	6.8
6	Bladder	440,864	4.7
7	Esophagus	418,350	4.5
8	Non-Hodgkin lymphoma	304,151	3.3
9	Kidney	271,249	2.9
10	Leukemia	269,503	2.9

Top 10 cancers in women:

Rank	*Cancer*	*New cases in 2020*	*% of all cancers*
All	cancers*	8,751,759	--
1	Breast	2,261,419	25.8
2	Colorectal**	865,630	9.9
3	Lung	770,828	8.8
4	Cervix uteri	604,127	6.9
5	Thyroid	448,915	5.1
6	Corpus uteri	417,367	4.8
7	Stomach	369,580	4.2
8	Ovary	313,959	3.6
9	Liver	273,357	3.1
10	Non-Hodgkin lymphoma	240,201	2.7

Data Preprocessing

The downloaded articles were preprocessed to remove any irrelevant information, such as headers and footers. Text normalization techniques were applied to standardize the text format, ensuring consistency throughout the dataset. We also converted the txt files to excel and combined them.

Tokenization

Tokenization was performed to break down the text into individual words or phrases. This process allows for a more granular analysis of the content and helps capture the underlying semantic meaning of the articles. We normalized the tokenized dataset and saved it as normalized corpus for the clean article column in our final data frame.

Vectorization

The tokenized text data was transformed into a numerical format suitable for machine learning algorithms. We employed vectorization techniques TF-IDF (Term Frequency-Inverse Document Frequency), to represent the textual data as numerical vectors. We ended up with approximately 90,000 features for our final model.

Classification Algorithms

To classify and predict the cancer types based on the article content, we employed the following machine learning classifiers:

Naive Bayes

Naive Bayes is a probabilistic classifier that applies Bayes' theorem with the assumption of independence among features. It calculates the probability of each class given the input data and selects the class with the highest probability.

Logistic Regression

Logistic Regression is a linear model that uses a logistic function to model the probability of a certain class. It estimates the parameters that best fit the data and makes predictions based on the calculated probabilities.

Support Vector Machines (SVM)

SVM is a powerful classifier that finds an optimal hyperplane to separate the data points into different classes. It maps the input data to a high-dimensional feature space and maximizes the margin between classes.

Stochastic Gradient Descent (SGD)

SGD is an optimization algorithm commonly used for training linear classifiers. It updates the model's parameters in each iteration, considering only a small subset of the training data, making it suitable for large-scale datasets.

Random Forests

Random Forests is an ensemble method that combines multiple decision trees to make predictions. It constructs each tree using a random subset of the training data and features, reducing the risk of overfitting.

Model Evaluation

We evaluated the performance of each classifier using appropriate evaluation metrics, such as accuracy, precision, recall, and F1 score. Cross-validation techniques, such as k-fold cross-validation, were employed to obtain reliable and unbiased performance estimates. We also did outside testing with a randomly sampled 10,000 datasets for each of the categories like men, women, and mixed.

Results and Discussion

In this section, we will be looking at the performances of the various models for both inside and outside testing. We chose the best performing model and we hyper tune it for outside testing once again.

Mixed

Mixed dataset inside testing results, give us very high performances. Especially Linear SVM. We did cross validation 5 -fold. We did multiple tests on 90k, 50k, 30k, and 10k datasets. Inside testing results:

	Model	CV Score (TF)	Test Score (TF)
0	Naive Bayes	0.911906	0.906000
1	Logistic Regression	0.984469	0.984250
2	Linear SVM	0.985781	0.985125
3	Linear SVM (SGD)	0.985781	0.986250
4	Random Forest	0.905813	0.908750

We sampled ten thousand articles for outside testing earlier in the pipeline. We preprocessed the outside testing data set using the same techniques and vectorized the dataset using the same TF-IDF object. We conducted another test using the outside testing data on our trained models:

	Model	CV Score (TF)	Test Score (TF)
0	Naive Bayes	0.911906	0.908000
1	Logistic Regression	0.984469	0.981875
2	Linear SVM	0.985781	0.983125

3	**Linear SVM (SGD)**	**0.985781**	**0.982500**
4	**Random Forest**	**0.905813**	**0.912750**

From the combined results, we chose Linear SVM as a robust model that can perform very well with the given data set. Here are the reports for the hyperparameter tuned model:

Inside Test Report
Accuracy: 0.9878
Precision: 0.9877
Recall: 0.9878
F1 Score: 0.9877

Outside Test Report
Accuracy: 0.9851
Precision: 0.9852
Recall: 0.9851
F1 Score: 0.9851

Finally, we have a classification report for the hyper tuned Linear SVM model performance of outside testing data:

	precision	recall	f1-score	support
Stomach	**0.97**	**0.96**	**0.97**	**775**
Lung	**0.99**	**1.00**	**0.99**	**805**
Bladder	**1.00**	**1.00**	**1.00**	**802**
Oesophagus	**0.97**	**0.97**	**0.97**	**827**
Cervixuteri	**1.00**	**0.99**	**0.99**	**765**
Liver	**0.98**	**0.98**	**0.98**	**803**
Thyroid	**1.00**	**1.00**	**1.00**	**789**
Breast	**0.99**	**1.00**	**0.99**	**839**
Prostate	**1.00**	**1.00**	**1.00**	**825**
Colorectal	**0.98**	**0.98**	**0.98**	**770**
accuracy			**0.99**	**8000**
macro avg	**0.99**	**0.99**	**0.99**	**8000**
weighted avg	**0.99**	**0.99**	**0.99**	**8000**

Men

The Men's dataset performed similarly with great results for cross validation and outside testing. Inside testing results:

	Model	CV Score (TF)	Test Score (TF)
0	Naive Bayes	0.905969	0.903375
1	Logistic Regression	0.975781	0.977625
2	Linear SVM	0.979375	0.979625
3	Linear SVM (SGD)	0.978281	0.979250
4	Random Forest	0.893719	0.883375

Outside testing results with the 10k testing dataset extracted earlier in the pipeline:

	Model	CV Score (TF)	Test Score (TF)
0	Naive Bayes	0.905969	0.905125
1	Logistic Regression	0.975781	0.976000
2	Linear SVM	0.979375	0.979250
3	Linear SVM (SGD)	0.978281	0.977625
4	Random Forest	0.893719	0.880625

Hyperparameter tuned Linear SVC model performance:

Inside Test Report
Accuracy: 0.9814
Precision: 0.9814
Recall: 0.9814
F1 Score: 0.9814

Outside Test Report
Accuracy: 0.98
Precision: 0.9801
Recall: 0.98
F1 Score: 0.98

Finally, we have a classification report for the hyper tuned Linear SVM

model performance of outside testing data:

	precision	recall	f1-score	support
Oesophagus	0.98	0.97	0.97	765
Prostate	1.00	1.00	1.00	805
Liver	0.98	0.98	0.98	775
Lung	0.99	1.00	0.99	839
Bladder	0.98	1.00	0.99	803
Stomach	0.96	0.96	0.96	825
Leukemia	0.97	0.99	0.98	802
Colorectal	0.98	0.98	0.98	770
Kidney	0.99	0.97	0.98	789
Non-Hodgkin	0.99	0.97	0.98	827
accuracy			0.98	8000
macro avg	0.98	0.98	0.98	8000
weighted avg	0.98	0.98	0.98	8000

Women

The women's dataset and its inside testing results, the performance was, as with the others, exceptionally high, with Linear SVM (SGD). The Inside testing results are as follows:

Model	CV Score (TF)	Test Score (TF)
0	Naive Bayes	0.880375	0.870750
1	Logistic Regression	0.971594	0.972000
2	Linear SVM	0.973906	0.971875
3	Linear SVM (SGD)	0.973937	0.973000
4	Random Forest	0.859938	0.875125

The corresponding outside testing results with the extracted 10k dataset from the original 100k dataset.

	Model	CV Score (TF)	Test Score (TF)
0	Naive Bayes	0.880375	0.886125
1	Logistic Regression	0.971594	0.971125
2	Linear SVM	0.973906	0.972750
3	Linear SVM (SGD)	0.973937	0.973375
4	Random Forest	0.859938	0.880625

Hyperparameter tuned Linear SVC model performance:

Inside Test Report
Accuracy: 0.9754
Precision: 0.9754
Recall: 0.9754
F1 Score: 0.9754

Outside Test Report
Accuracy: 0.9755

Precision: 0.9756
Recall: 0.9755
F1 Score: 0.9754

And lastly, we have a classification report for the hyper tuned Linear SVM model performance of outside testing data:

	precision	recall	f1-score	support
Colorectal	0.96	0.98	0.97	770
Endometria	0.98	0.98	0.98	825
Ovarian	0.98	0.98	0.98	775
Lung	0.99	1.00	0.99	803
Thyroid	0.98	0.99	0.99	765
Non-Hodgkins Lymphoma	0.96	0.95	0.96	802
Breast	0.98	0.98	0.98	839
Stomach	0.98	0.97	0.97	827
Liver	0.94	0.94	0.94	789
Cervical	0.98	0.98	0.98	805
accuracy			0.98	8000
macro avg	0.98	0.98	0.98	8000
weighted avg	0.98	0.98	0.98	8000

Dataset Size Robustness Testing

To ensure we have the optimal dataset size for the model. We did experiments with different dataset sizes. We used the same models throughout. Additionally, we experimented with different vectorization methods.

90,000 Mixed Dataset with CounVectorizer results:

	CV Score (TF)	Test Score (TF)
Naive Bayes	**0.875403**	**0.878333**
Logistic Regression	**0.959611**	**0.954389**
Linear SVM	**0.955778**	**0.951778**
Linear SVM (SGD)	**0.956778**	**0.954222**
Random Forest	**0.844958**	**0.850778**
GBM	**0.962458**	**0.963056**

50,000 Mixed Dataset with CountVectorizer results:

	CV Score (TF)	Test Score (TF)
Naive Bayes	**0.872235**	**0.879321**
Logistic Regression	**0.960374**	**0.961288**
Linear SVM	**0.955486**	**0.958621**
Linear SVM (SGD)	**0.954177**	**0.956745**
Random Forest	**0.861347**	**0.850879**
GBM	**0.961757**	**0.962769**

40,000 Mixed Dataset with Count Vectorizer:

	CV Score (TF)	Test Score (TF)
Naive Bayes	**0.870821**	**0.875682**
Logistic Regression	**0.961493**	**0.961742**
Linear SVM	**0.958209**	**0.957121**
Linear SVM (SGD)	**0.955075**	**0.957727**
Random Forest	**0.86403**	**0.897727**

GBM	**0.962164**	**0.96303**

40,000 Mixed Dataset with TF-IDF:

	CV Score (TF)	Test Score (TF)
Naive Bayes	**0.8645312**	**0.869125**
Logistic Regression	**0.96375**	**0.9685**
Linear SVM	**0.96325**	**0.967625**
Linear SVM (SGD)	**0.965468**	**0.96725**
Random Forest	**0.8626875**	**0.86925**

30,000 Mixed Dataset with CountVectorizer:

	CV Score (TF)	Test Score (TF)
Naive Bayes	**0.870498**	**0.870505**
Logistic Regression	**0.960746**	**0.961111**
Linear SVM	**0.956468**	**0.956667**
Linear SVM (SGD)	**0.952438**	**0.952222**
Random Forest	**0.87**	**0.875758**
Gradient Boosted Machines	**0.961144**	**0.963535**

30,000 Mixed Dataset with TF-IDF:

	CV Score (TF)	Test Score (TF)
Naive Bayes	**0.869452**	**0.868383**
Logistic Regression	**0.963482**	**0.963838**
Linear SVM	**0.9638308**	**0.9640404**
Linear SVM (SGD)	**0.9653731**	**0.966667**
Random Forest	**0.8659203**	**0.880505**

Discussions

Which classifier do you prefer?

The Linear SVM classifier is preferred among the classifiers used in this project. This choice is based on its consistently high performance across all datasets (men, women, and mixed) in terms of accuracy, precision, recall, and F1 scores. The Linear SVM model demonstrated robustness and generalizability in both inside and outside testing scenarios.

The number of tokens? (Feature Weighting and Selection?)

Regarding the number of tokens, the dataset contained approximately 90,000 features after tokenization and vectorization using the TF-IDF technique. TF-IDF represents the importance of each term (token) in the dataset by calculating the term's frequency in a document and its inverse document frequency across the entire corpus. This allows for effective feature weighting, as the more significant and informative terms receive higher weights.

Training Time and Testing Time?

In terms of training and testing time, the performance can vary depending on the size of the dataset and the complexity of the chosen classifier. However, SVM models, including Linear SVM, generally have relatively faster training times compared to other complex models like Random Forests. SVMs optimize a convex objective function, which makes

training faster. Testing time can be efficient as well, especially with optimized models and smaller feature sets.

Individual Contributions

Our group members are Amina Enkhbayar, Anand Enkhbayar, Yu Chun Lin. Given the scale of the project we decided to divide the project into 3 stages. Collect the data and clean. Train the model and evaluate. Find the best model, hyper tune, and outside test. We wanted each member to have experience going through the steps of the NLP pipeline, so we divided the work by gender categories. Male dataset was given to Anand, female dataset was given to Yu Chun Lin, and the mixed dataset was given to Amina. We collaborated on the pipelines structure and shared resources but the results for each category were up to the group members' abilities.

After we created a pipeline that produced desired results, we compiled our notebooks and converted them into one format so that we can reproduce our results multiple times with different dataset sizes and parameters.

We wanted to run experiments for different dataset sizes for the mixed dataset category. Therefore, we divided the dataset sizes and ran our experiments.

Amina tested the dataset sizes ranging from 10k to 30k; **10k mixed TF-IDF, 10k mixed CV, 15k TF-IDF, 15k CV, 30k mixed TF-IDF, 30k mixed CV.**

Anand tested the dataset sizes: **90k mixed TF-IDF, 90k mixed CV, 50k TF-IDF, 45k TF-IDF.**

Enid tested the dataset sizes: **40k mixed TF-IDF, 40k mixed CV plus experiments with stratified sampling.**

Having a medical student on our team has helped us understand our datasets better. Enid gave us great insight into the meaning of the abstracts and the medical terminology. Having a domain expert helped us concluded that TF-IDF was a better suited vectorization method for our pipeline.

In conclusion, the workload was very balanced with each member being responsible for 33.33% of the workload.

Personal Comments

Anand: I believe the final project was a great challenge for us. But it was also a great opportunity for us to learn more about machine learning. I am now more aware of the resource constraints associated with running bulky large datasets. When we were experimenting with the largest data sets of 90,000 articles, the gradient boosting machine took 3-4 hours to train the model. Hyperparameter tuning took a further 3 hours. In the end we had no choice but to drop the classifier considering the running time and accuracy scores. Making sure that the data is preprocessed in a precise way that does not lose crucial semantic information about the relationships was very important.

Amina: It was a great experience. Creating an efficient and streamlined machine learning pipeline helped us conduct multiple tests quickly. So, we can see the direct results of our model choice, parameter choice, and data quality. We had the opportunity to experiment with different dataset sizes and different vectorization and normalization techniques. The notebooks that were provided by the professor helped us efficiently process the data.

Enid: In the context of interdisciplinary collaboration, professor provided us with ample opportunity to collaborate and learn from each other. As a medical undergraduate student, the guidance from the professor has helped me gain a clearer understanding of how we can effectively utilize data.

Conclusion

In conclusion, our analysis utilized natural language processing (NLP) techniques and machine learning algorithms to classify and predict cancer types based on scientific article content. We collected a dataset of 300,000 articles on the top 10 most frequent cancer types in men, women, and mixed populations.

After preprocessing the data, including text normalization and tokenization, we employed vectorization techniques such as TF-IDF to transform the textual data into numerical vectors. We evaluated the performance of various classifiers including Naive Bayes, Logistic Regression, Support Vector Machines (SVM), Stochastic Gradient Descent (SGD), and Random Forests.

For each dataset (men, women, and mixed), we conducted both inside testing and outside testing. The models consistently performed well in both cases, with Linear SVM demonstrating the highest accuracy and precision scores across all datasets.

Furthermore, we hyper-tuned the Linear SVM model and achieved even better results. The hyper-tuned model showed high accuracy, precision, recall, and F1 scores for both inside and outside testing datasets.

Overall, this analysis showcases the effectiveness of machine learning and NLP in classifying cancer types based on scientific article content. The findings can contribute to improved diagnostic and treatment strategies in the field of cancer research.

YouTube Video Link: https://youtu.be/bSbJl8iB7jQ