Laylani Callaway, Casey Devine, Tyler March, Jasper Sylvestre, Lucy Liu
Dr. Sengupta
ST 495
12/09/2023

Final Report

Abstract

The National Cancer Institute says that approximately 39.5% of men and women will be diagnosed with cancer at some point during their lifetimes [5]. That number is increasing, making it more important than ever to have reliable cancer treatment for patients. Radiation Oncology is one of the most popular and effective means of treating cancer patients and requires many experienced professionals utilizing a series of very complex processes, systems, and devices. This makes the treatment process extremely susceptible to errors. As it currently stands, the classification of errors in this process requires a medical professional to parse through free text descriptions of medical errors and manually come to informed conclusions on different features of the event. A report published by the National Library of Medicine states that less than 10% of medical errors are reported [4]. We aim to use Natural Language Processing (NLP) and supervised statistical learning classification methods to classify deviation events into discrete categories, streamlining the reporting process and relieving human reporters from the burden of manual categorization. The eventual goal is to use this data to predict future deviation events and minimize patient risks. Our resulting models showed some imperfect but promising results in classifying a sample of the deviation events into broad categories.

Introduction

Medical error is one of the leading causes of death in the United States. Finding ways to mitigate this issue is crucial to improving health care and lowering economic impacts [1]. This paper focuses on the usage of Natural Language Processing (NLP) and supervised statistical learning classification methods to identify patterns in medical error reports and further reduce errors in radiation oncology. Medical personnel often report medical errors through a description of the error. Because of this, there is not a standardized way of reporting these errors. Many error types exist, with some extremely serious treatment-related errors and others simple administrative errors. NLP can be used to break down these free text descriptions into numerical data, allowing us to implement statistical analysis on said data. This will help track and standardize the reporting of these errors, which can help determine how to prevent repeating the same medical mistakes [1].

In 2022, there were 1.9 million new cancer cases and over 600,000 cancer deaths [2]. At least half of cancer patients receive radiation therapy as a part of their treatment. We focus on radiation oncology because it is susceptible to error and affects many lives worldwide. Radiation requires multiple systems, healthcare workers, and devices, often leading to mistakes. We aim to use data and classification methods to predict error categories. NLP and statistical learning methods will simplify error classification more than sifting through text and lengthy descriptions. NLP also eliminates human bias and misinterpretation when classifying error types that may otherwise be inconsistently grouped [1].

The future goal for this project is to create a system for radiation oncologists and other medical professionals that will use machine learning methods to help flag potential future errors in the radiation oncology process before they occur. As radiation oncology continues to become more widely used in patient treatment, it will be important to ensure that it continues to be a safe treatment, and a tool that can flag potential errors will be essential to that goal.

Data Description

For this assignment, we are using data collected from the USLCC test center's database provided to us by Ed Kline from RadPhysics Services LLC to attempt to perform statistical analysis. This data consisted of individual instances of error reports from events that occurred post-treatment. The primary dataset consisted of two main variables of interest: one representing the error code and one representing a free text description of the error event as inputted by the person who reported the event. The error code represented a four-tier classification of the error event in the form of "<pre or post treatment>/<error category>/<error subcategory>/<error attribute>". These are our main two fields because we are attempting to create a classification method that will allow us to categorize an error event into as narrow a category as possible using the free text description included in the error report.

In addition to the two main variables of interest, our raw data also included variables for the username of the person who inputted the error, the date on which the error was inputted, the intent of the treatment where the error occurred (curative or palliative), the method of the treatment and the type of error; all of which we intend to keep in our final data for the purpose of prediction. The data also contained a unique ID for the specific instance of the error, which we will keep to identify errors and many other variables that were not included in our final dataset as they were not deemed important for estimation either because the data was irrelevant or too sparse. We also pulled data from lookup tables provided with the raw data to include the job role of the person who inputted the error for use in estimation and the text description of each level of the error classification hierarchy for ease of interpretation.

When cleaning our free text field, we removed the text of any punctuation and lowercase all of the text. The idea is to ensure that when we perform any language processing, there are fewer discrepancies between two instances of the same word for the computer to recognize them as equivalent.

To get a variable that we can use as a response, we broke the error code up into new variables that represent the category of the error at each level of the categorization hierarchy (e.g., "Quality Assurance", "Quality Assurance>Checks", "Quality Assurance>Checks>Physics sign-off/approval of linac fault log miss./late"). We can first build models to attempt to estimate categorization at the most specific levels and reduce specificity until we get an "acceptable" error rate in our models.

With the current state of our data, we also expect it to be unrealistic to create classification models that will predict "Event Category" (the second broadest categorization level) with any acceptable error rate. We believe that predicting "Pre/Post treatment" (the broadest level of categorization) is not useful enough of a result and is impossible, considering all of our data is post-treatment. In response, we created a level of categorization that will reside between those two levels of specificity that we expect to be realistic and informative. We call this the "Broad Category", representing the step of the clinical workflow in which the error occurred. We created this categorization by collapsing the Event Category into four supercategories described in "Automated Error Labeling in Radiation Oncology via Statistical Natural Language Processing."

Broad Category was composed of four distinct levels: "Administrative", "Standards", "Treatment", and "Treatment Preparation"; the collapse of the categories into these supercategories can be seen in **Figure 1** below.

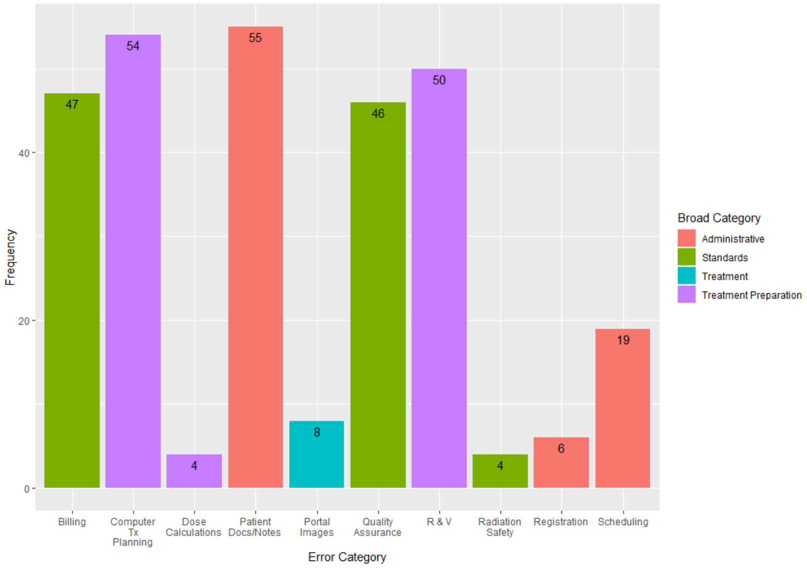

Figure 1. Frequency distribution of the 10 event categories (**bottom**) and the 4 broad categories (**right**).

This collapsing of classification levels succeeded in improving the sizes of most of the response categories to sizes large enough for statistical analysis, with the one exception of the "Portal Images" category being a one-to-one match with the "Treatment" supercategory and that frequency still being too small to provide any predictive power to our model.

After presenting this data and our initial classification error rates to Dr. Sengupta, he suggested that we sacrifice being able to predict the "Treatment" category for the increase in predictive power that would result from its removal from our models altogether. As a result, the

“Treatment” observations were removed from our final dataset, bringing it from 293 records down to 285 and leaving us with three possible response levels; the new frequency distribution is shown in Figure 2 below.

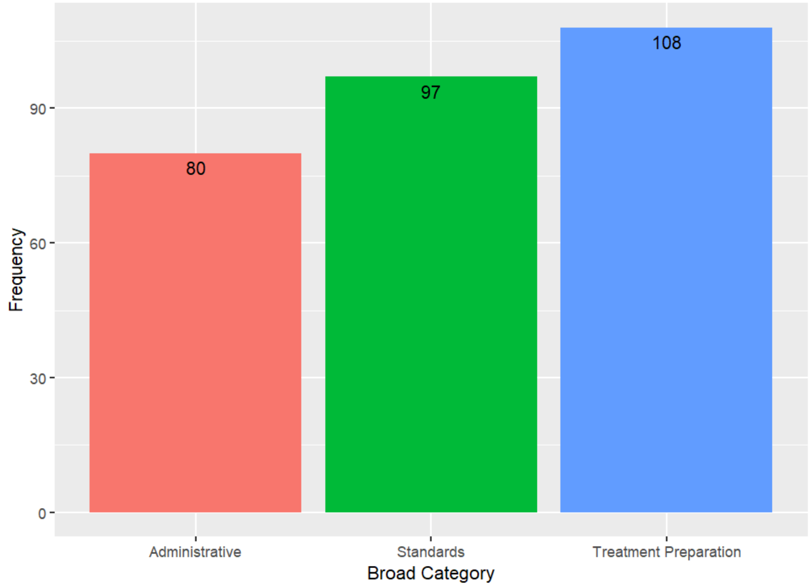

Figure 2. Frequency distribution of the 3 remaining event categories.

After all of this manipulation of the data, we concluded that the only subset of the original data set that would be able to be used for predictive analysis would be this new dataset represented in Figure 2; We removed the other response options from the cleaned dataset and made the final change of completely removing the rows that corresponded to the “Treatment” broad category level. This left us with a dataset of 285 observations and 11 total variables made up of one ID variable for identifying the individual error instance, one BroadCat variable to act as the categorical response for the data, and nine categorical variables to be used as predictors in estimating the response; more detailed information on these variables can be found in the “USLCC Error Chart.html” help file that will be included with the cleaned dataset “USLCC Error Chart.csv” in our final delivery.

Methods

To tackle this problem, we decided to split the entire roster of supervised statistical learning classification methods at our disposal amongst the members of this group so everyone could focus their attention on a subset of the models. This allowed us to compare all of our strategies and error rates before making any final decisions on statistical methods.

Support Vector Machines

Support Vector Machines (SVMs) are a complex supervised statistical learning method that focuses on the boundary points between classification levels of the data to form multidimensional hyperplanes to serve as boundaries between these classification levels in the data.

The approach to SVM taken was first to read the data set into R, then impute the missing "IDbyRole" values with the administrator code "ABDEGKMNPRTU", then to code all categorical variables in the data set beside the free text description as a factor. Next, 50 iterations of cross-validation were performed, with 80% of the dataset used as training data and the remaining used as testing data. Within the cross-validation, each data set was converted to a term document matrix, the stop words were removed, and a maximum sparsity cap of 80% was enforced. Then, Latent Semantic Analysis was performed to extract numerical features from the free text fields in the dataset. These extracted features were scaled and added back into their respective datasets. An SVM model was then fitted on the training dataset utilizing all of the numerical and categorical features with the response of BroadCat. The model was used to predict the classification levels of the test data, and the overall classification error rate was recorded and the summary statistics are included in Table 1 in the results section.

Tree-Based Methods

Both classification trees and random forest models in this context are types of supervised models where the response variable is a categorical variable. Classification trees are hierarchical structures where each branch represents a feature being split by a rule and eventually end at leaf nodes that represent the decided classification label. Classification trees are prone to overfitting and lack the robust nature of random forest models which in return are usually less interpretable. Random forest models use an ensemble of multiple decision trees. Each tree is trained on a subset of sampled data and a sampled subset of features are used at each node. Due to the advantages random forest models offer, they were used here.

The approach taken here was to read the data set into R, filter out treatment values of the broad category response variable, remove the ID variable, the category variable, the subcategory variable, the attribute variable, and the date variable. The date variable was converted into POSIX time as "TimeSeconds" to be usable as a numeric variable. All variables but the description and the construct time variable were set to factor variables.

The data was split with 80% being used for training data and 20% for testing data. 50 cross-validation iterations were used to attempt to gather a mean cross-validation classification error score close to the true mean. Within cross-validation, the constructed time variable was scaled using the testing data's time variable's mean and standard deviation values to ensure there was no data leakage. The same was done to fill in missing values of the "IDbyRole" variable, except the training data's "IDbyRole"'s mode was used instead. Latent semantic analysis was also performed at this time and is done exactly the same as is described above in the section on SVM.

A random forest model was trained with and without the data extracted from the LSA. *m* represents the number being randomly sampled at each split for the model. The model using LSA had an *m* value of 4 and the model without LSA data had an *m* value of 3. 500 trees were used for both models. Broad category was treated as the response variable with all the other variables given as predictors, and the models were both constructed within each iteration, they were then used to predict the broad category values, and the classification error rate was calculated for each iteration for both models. The summary statistics are for the classification error rates for both models are shown in the results section in Table 1. The 95% confidence intervals for the true mean classification error rate for both models are shown in the results section in Table 2.

K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm is a supervised machine learning method used for classification and regression tasks. It operates on the principle that similar instances in a feature space tend to belong to the same class. The primary goal of employing KNN in our research is to predict the categorical variable BroadCat based on the features derived from the dataset. The algorithm's performance is highly dependent on the choice of the number of neighbors (k), making it crucial to explore a range of k values and identify the one that minimizes prediction errors. Through this approach, we aim to develop an effective predictive model for categorizing instances in our dataset.

To start, we loaded the necessary tools and imported our dataset, making sure to focus on the levels equally by filtering out instances labeled as "Treatment." Moving on to the core of analysis, Natural Language Processing (NLP) techniques broke down the text into meaningful parts and extracted features. In this stage of our analysis, we engaged in text processing and feature extraction, with a specific focus on employing Latent Semantic Analysis (LSA). This involved converting the free-text data into a structured text corpus, recognizing that the small sample size led to unique text units. Notably, the resulting term-document matrix exhibited sparsity, characterized by an abundance of zeros, particularly in cases where singular values dominated due to the limited dataset size. The sparse nature of the matrix emphasized the need for careful consideration in subsequent analyses, given the prevalence of zero values and the potential impact on the interpretability of the results.

Next, the dataset is divided into 80% training and 20% testing sets, iterating 50 times, for training KNN models, a common practice in machine learning. To find the best value of 'k' (the number of neighbors to consider), performance is cross-validated, setting the range 1 to 50 folds. The goal was to find the optimal 'k' that minimized errors. The output shows the cross-validation error is minimum when k = 2. The classification error rates were recorded and are summarized in Table 1 in the results section.

Post-training, the model performance is evaluated using a confusion matrix and associated metrics. These metrics helped us assess the model's ability to distinguish between different error categories. 95% Confidence Interval for the mean classification error is also calculated and shown in the later section.

Discriminant Analysis / Naive Bayes

Linear discriminant analysis (LDA) is a dimensionality reduction method used for classification. Quadratic discriminant analysis (QDA) is similar but allows for non-linear boundaries separating groups and non-constant variances. Naive Bayes (NB) involves Bayes' theorem. This popular probabilistic theorem uses the probability of event A given the occurrence of event B to find the probability of event B given event A. NB also assumes that predictors are independent within each class.

We attempted both types of discriminant analysis. However, we continued without using the QDA model because we ran into a grouping size error. We determined that with the time we had it would be more beneficial to further evaluate the other classification methods that yielded better results.

All categorical variables and missing values were handled in a method consistent with that used for SVM. For both LDA and NB, we used 50 iterations of cross-validation and a sparsity limit of 80%. The data was split into 70% training and 30% testing data. We also used LSA on the free text description. Interestingly, we found that LDA and NB models using only numeric variables had lower classification error rates than those using the text data. The interpretation of these results is unclear, so we omitted them from this report. In the future, models without LSA should be considered for more analysis because at least three of the classification methods we used reported similar results.

The error rates for these two models are relatively high. It is debatable which of these two models performed better because LDA yielded a lower minimum error rate, but the median and maximum error rates are lower for NB. We decided to look further into the NB model because the error rates were lower on average. As shown below in Figure 5, this model performed well when predicting the standards category but otherwise needs improvement.

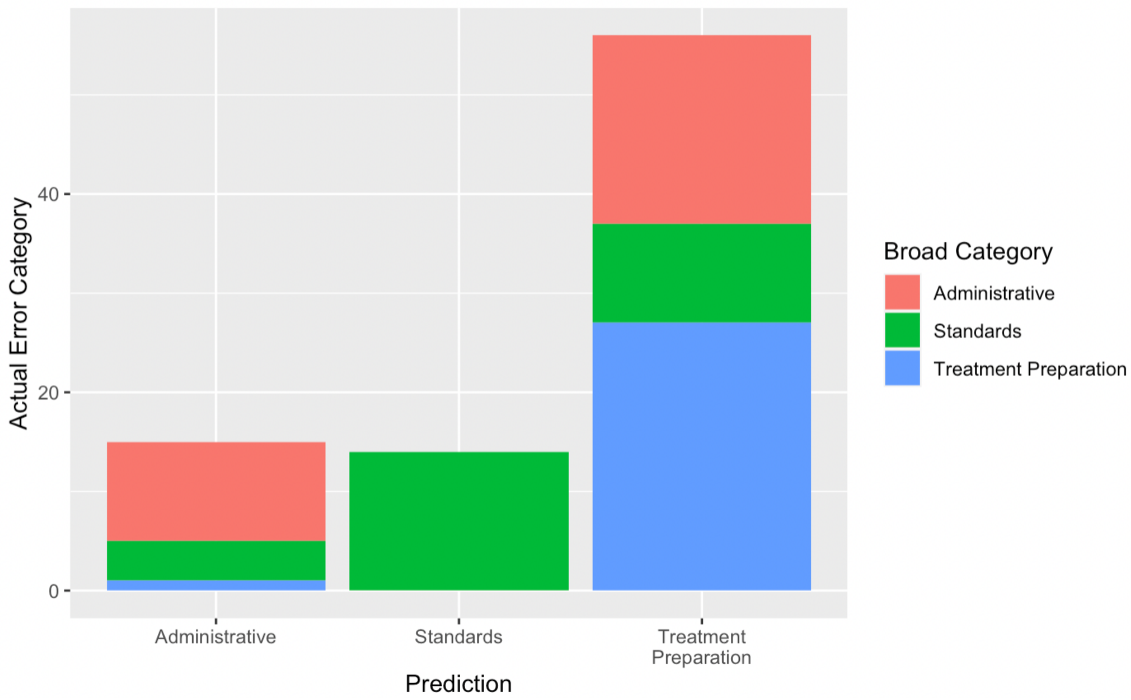

Figure 3. Stacked barplot showing the predicted versus actual categories for the NB model

Challenges

We ran into challenges in our assignment's data interpretation/cleaning and modeling portions.

Data

The first challenge we ran up against was interpreting the large amount of data given to us from the MERP database for this assignment. The cut of data given to us included 15 Excel spreadsheets full of various types of data that were hard to interpret since we did not have any background on the data, and there was not any explicit guide to what the data meant. This was rectified by a meeting with Ed Kline, where we were given the general rundown of the datasets of focus and the desired response variable. This gave us a starting point on what to look for in the data and led to us eventually having a good understanding of the dataset.

The next data challenge we ran into was the format of the cleaned dataset, specifically regarding the response variable. We knew that we wanted to be able to test for different levels of specificity in the response, but we were unsure of the best way to do so. There was indecision between duplicates of each row where the response corresponded to a different category level where we would have to subset the rows during modeling and have different variables representing different response options. We would have to filter the columns during modeling. Both of these approaches had pros and cons, but eventually, the group decided on the multiple-column approach as it would be easier to approach modeling with the dataset in this format.

Modeling

The biggest challenge in modeling was getting models that could predict the response for every category level with sufficient power. The biggest reason for this was because of the low frequency of records in the “Treatment” level of the broad category; with Dr. Sengupta’s help, we were able to decide to sacrifice these records from our model entirely to improve our predictive power for the remaining levels of the response.

Another significant challenge was performing Latent Semantic Analysis (LSA) on the text data, as this is a relatively new concept for the group, and took a little bit of working with it to figure it out. It was also difficult to figure out how to use LSA to its full potential and to ensure that it was helping our analysis and not hurting it. We eventually benefitted the most from LSA when we instilled a maximum sparsity and removed stop words.

We also had challenges with running the data through specific models. We were planning on creating a logistic regression model to analyze how the descriptions can classify the broad categories, but ran into problems when we realized that the response variable was not binary. We understood how to use R code to run logistic regression when the output is binary with the “glm” function, but were unsure what to do with non-binary responses. We figured that this model would not be as accurate as the other kind of models anyway, so we decided to not use logistic regression and move forward with other models. As previously stated, there was an issue with the group sizes when attempting to model using quadratic discriminant analysis. We felt that we had a sufficient number of classification models without using QDA. Given more time, we would investigate this issue in more depth.

Results

The following shows the results of the classification statistical methods.

Classification Method	Min.	Q1	Median	Mean	Q3	Max.
Random Forests with LSA	3.51%	7.02%	10.53%	11.75%	14.04%	31.58%
Random Forests without LSA	1.75%	5.26%	7.90%	8.21%	10.52%	17.54%
Support Vector Machines	3.51%	8.77%	10.53%	13.61%	17.11%	31.58%
K-Nearest Neighbors	47.37%	57.02%	60.53%	60.18%	64.91%	66.67%
Linear Discriminant Analysis	21.18%	44.41%	58.82%	54.78%	65.59%	74.12%
Naive Bayes	32.94%	40.29%	47.06%	47.81%	54.12%	64.71%

Table 1. Summary statistics of the classification error rate by Classification Method

Classification Method	Mean	95% Confidence Interval for mean error rate
Random Forests with LSA	11.75%	(10.03%, 13.48%)
Random Forests without LSA	8.21%	(7.20%, 9.23%)
Support Vector Machines	13.61%	(11.63%, 15.60%)
K-Nearest Neighbors	61.96%	(60.23%, 63.70%)
Linear Discriminant Analysis	54.78%	(50.90%, 58.65%)
Naive Bayes	47.81%	(45.58%, 50.04%)

Table 2. 95% Confidence Intervals for mean classification error rate by Classification Method

Discussion

With how prone to errors that the field of oncology can be, there needs to be methods to detect errors so that they can be prevented in the future. With medical errors being one of the leading causes of death especially in this field, finding a way to detect these errors would be a big step in limiting these errors.

In our study looking into how NLP algorithms can be used to automate the tracking of errors, we found that there are models that can perform well even with small sample sizes. We found that with this data, we can use classification models to detect broad categories from the descriptions that were given in the training datasets. With the amount of time that we had, we were unable to look deeper into every aspect of possibilities of what we could do. Going deeper into using more features of what LSA can offer us could improve the accuracy of the models. We can use these methods to maybe predict error categories and go deeper than the broad categories.

This could further support the research of implementing Explainable Artificial Intelligence (XAI) into similar settings. The way XAI can look past contextual bias and observe larger patterns, these models could greatly improve the workflow and error reporting in radiation oncology. The models that we made were able to broad categories, but there are definitely flaws that can be made, such as misclassifying treatment with treatment preparation. That is why we ended up dropping the treatment category. There is no way of seeing if that could be happening in other categories. We do not advocate completely replacing humans with artificial intelligence that detects and reports deviations automatically. However using this basis and having human controls could be very successful in the future of detecting errors in oncology. The strategy that we suggest is to use artificial intelligence to help humans observe actions of human reports and make suggestions when the patterns deviate from the expectations. This type of AI will keep control in human hands and improve interpretability while assisting them by giving access to experiences beyond reporter prospective. This will also reduce instances where the same error is written different ways, providing clear differences and patterns that can be easily tracked with models.

Going forward, we hope to see predictive models be embedded into automated detection systems to provide assistance for radiation therapy teams. Models, such as the ones that we developed, is a good place to start in further advancing the field of oncology. It takes away the bias that humans can cause when detecting errors. This can help eliminate blame culture when looking at medical errors and really start helping people learn from mistakes, so that they do not happen in the future. We recommend that we use the principles that we mentioned in our models and what we discussed in this section to really innovate the way oncology works for the better.

Conclusions

In summary, our study explored the intricate domain of radiation oncology, addressing the crucial need for error detection and prevention. By implementing Natural Language Processing (NLP) in the form of latent semantic analysis alongside various classification methods, our goal was to simplify error reporting and relieve healthcare professionals by automating categorization. Notably, Support Vector Machines and Random Forests exhibited commendable performance.

Our study reveals the potential of NLP-aided statistical learning methods in this field, with models displaying adequacy in predicting broad error categories with a mean classification error rate between (10.03%, 13.48%) for random forest prediction and (11.63%, 15.60%) for SVM prediction despite the relatively small size of the data set. Acknowledging challenges like the elusive "Treatment" category, we hope to see further improvement in the predictive capability of these methods as more data becomes available.

Overall we see our results as extremely promising for the future of collaborative settings where AI aids human observers, minimizing biases, fostering a blame-free culture, and advancing patient safety through continuous learning. In essence, our study signifies a step towards innovating oncology practices, aligning technology with human understanding for a safer and more efficient future in radiation therapy.

Sources

1. Ganguly, Indrila, Graham Buhrman, Ed Kline, Seong K. Mun, and Srijan Sengupta. 2023. "Automated Error Labeling in Radiation Oncology via Statistical Natural Language Processing" *Diagnostics* 13, no. 7: 1215. https://doi.org/10.3390/diagnostics13071215

2. Siegel, RL, Miller, KD, Fuchs, HE, Jemal, A. Cancer statistics, 2022. CA Cancer J Clin. 2022. https://doi.org/10.3322/caac.21708

3. Data Provided by Ed Kline from RadPhysics Services LLC.

4. Anderson JG, Abrahamson K. Your Health Care May Kill You: Medical Errors. Stud Health Technol Inform. 2017;234:13-17. PMID: 28186008. https://pubmed.ncbi.nlm.nih.gov/28186008/

5. National Institute of Health. (2020, September 25). Cancer statistics. National Cancer Institute. https://www.cancer.gov/about-cancer/understanding/statistics