Predicting Lung Cancer Survival Times

03/28/2023

Eli Parker

Pooja Patel

Christopher Wilhite

Problem Statement and Background.

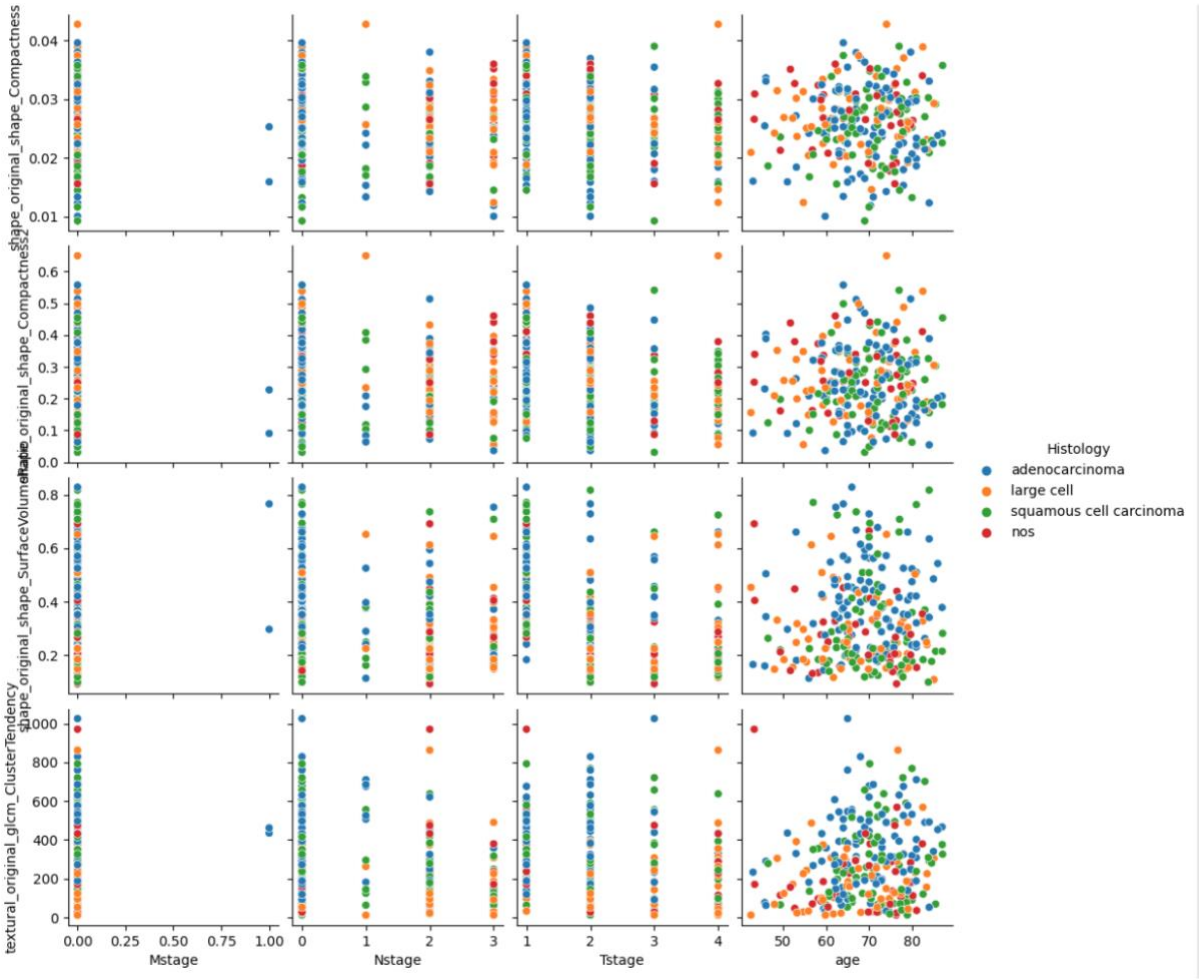
The data of our project originates from the National Institute of Cancer Imaging Archives, where real-life patients' CT scans have been taken to image their lungs. The data we are using is a conversion of the CT scans to a binary segmentation mask that is then used to create the numbers that represent the physical numbers. One of our informal success measures is to make sure the results from testing the models make logical sense. This project is beneficial for anyone who is or directly deals with a lung cancer patient. This has a large impact because it can be used to determine the best type of treatment through patterns allowing medical personnel to predict treatment reactions. Better solutions to the treatment of Lung Cancer could lead to a higher quality of life for the patients as well as a longer life expectancy. We are unaware of any related work and do not possess any external knowledge about lung cancer or the models used to predict survival times. However, we are actively researching the topic to further understand the variables used within our dataset.

Data and Exploratory Analysis

We were given a set of training data and testing datasets. Each of the datasets consisted of clinical data that held patients' personal information related to cancer and radiomics data that held patients' valuable information about cancer that cannot be identified with the naked eye. We have our datasets in two main CSV's: clinical_data and radiomics. The clinical_data CSV had one header row, while the radiomics CSV came with 3, which posed a problem to us that took multiple attempts to fix. To resolve this hurdle, we concatenated the three header rows together to make one. In order to clean the data, we started by dropping out any N/A's or data rows that

were left empty using a python script. We also deleted any rows, using a python script, where Tstage was greater than 4, Mstage was greater than 1, or Nstage was greater than 3.

After displaying the clinical data, we found that the Histology had two different types of not otherwise specified data. We also found some cases where the same data was represented in multiple datasets due to capitalization differences. We simply reinserted these values with the same capitalization. To understand the data, we took the now clean data and started by dropping the PatientID column to run summary statistics. Within the summary statistics, a lot of the characteristics go all over the place, so we were able to hand-pick some to correlate with each other. We made a pair plot with the seaborn library to make a sort of matrix of correlations between an x-axis of variables Mstage, Nstage, Tstage, age and a y-axis of variables shape_original_shape_Compactness1, shape_original_shape_Compactness2, shape_original_shape_SurfaceVolumeRatio, and textural_original_glcm_ClusterTendency.



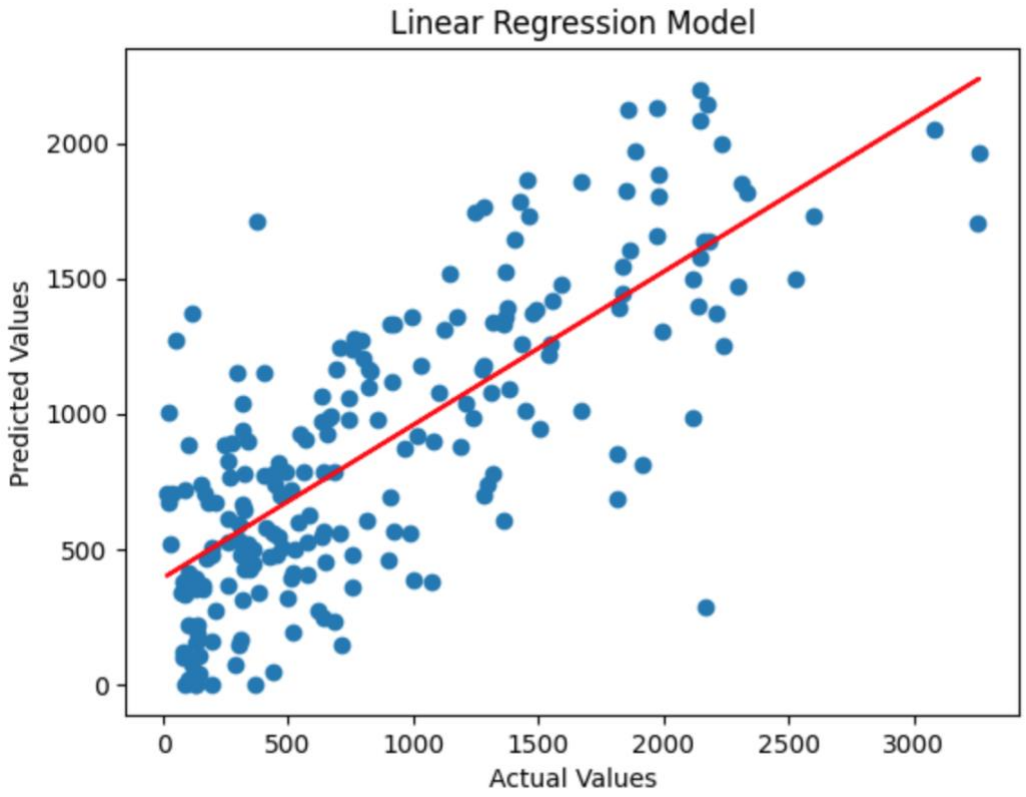
With these plots, the data seemed to be all over the place, and the –stage variables did not seem to give much insight. The age column in the pair plot gave clusters of points which we thought could lead to a meaningful correlation, so we made scatterplots with a line of best fit on each. Three of these plots had a flat line of best fit unfortunately, but the age vs. cluster tendency plot had a slightly positive slope, which could mean that the older the patient, the larger the clustering tendency could be. After this, we did our value cleaning for the Histology column and ran a histogram of the types of histology we have. The adenocarcinoma and squamous cell carcinoma histology seem to have more representation in our cleaned data. To avoid this bias, we can run statistics based on each type of histology.

Methods

The first model that we decided to use was Random Forest Regression because it utilizes simple decision trees based to sort the data into different branches. After creating the different branches, it can predict survival times by right censoring the data to make it align with the current patients' data. The model then takes the data it is given and compares it to the dataset within it to predict the righthand side that fits the data best or the survival time. This is a good method for predicting the survival time because it breaks the data up into a lot of different branches allowing it to do a very good job of comparing the given patient to the previously inputted data. Although it does a good job of sorting the data it does reduce the sample size greatly which reduces the accuracy of the result.

We also used Naive Linear Regression Model because it is a very basic model that assumes a linear relationship and estimates the coefficients of the linear equation to best fit the data. This implemented recursive feature elimination and features scaling with sklearn libraries. This model was a good model for providing a general case that lacked some accuracy as it is a best-fit model meaning that it simply took the given data points and creates an estimated line to best connect the data points. One issue with this model is that it includes several errors that come from the assumptions it makes. It assumes that the data set is linear, which is valid in many cases, and that any outliers or errors within the given data can be normalized for the line of best

fit to be created.



We are attempting the Kaplan-Meier because it estimates the survival time of the patients based on data from previous lung cancer patients. The Kaplan-Meier survival analysis uses probabilities for each patient compared to a data set that consists of the survival times for previous patients who have had lung cancer. The accuracy of this type of model depends on the amount of relevant data that exists for each patient with a certain cancer type. The accuracy of this model can be improved by restricting the base dataset to more closely match the specific patient, however, when doing this the base dataset becomes exponentially smaller decreasing the accuracy. This is a very good model for predicting survival times however no matter how closely the base dataset matches the specific patient there is a certain amount of error that cannot be eliminated.

When looking through modeling methods we decided to try the Cox regression model however we ran into some issues with not only the complexity of this model but also the fact that this model did not represent the data against time. The Cox regression model includes many different inputs that we did not have access to throughout this project such as patient treatment information like what type of treatment is being used and how the patient is reacting to the treatment. Due to the Cox regression model is an ever-changing prediction tool it doesn't do a good job of comparing the type of cancer to the life expectancy of the patient.

Tools

Our team developed this project in a Deepnote markdown file in python. We used a variety of python libraries to help put this together. To start off with the exploratory analysis, we used pandas to put our dataframes together, numpy to do operations on arrays, and matplotlib and seaborn to visualize the data. When we condensed the radiomics.csv header rows down, we used the "re" library to trim down the column names to make more sense. We also used libraries from sklearn and lifelines for our regression models. Lastly, we imported random to generate a random number for the random state number within the train and test splits to get varying results each time.

Our team communicated in a Microsoft Teams group. This proved to be an easy way to reach each other, also being what the university has people using for communication anyways. Throughout the process of this project, we have used Teams to delegate tasks and have done a good job of spreading around the workload among the team members.

Results

When implementing the Random Forest Regression model, the predicted values were all positive and all seemed to be within a reasonable range for survival times ranging from a few

hundred days to a few thousand. This suggests the model is at least producing results that are within the realm of possibility. However, due to the fact that the r2_score produced was closer to 0 than it was to 1, it indicates that the model was not able to capture much of the variation within the dataset and in turn was not a good fit for predicting survival times.

When implementing the feature elimination into the Naïve Linear regression model, we noticed that we need to keep the number of selected features to be as big as possible since when we lowered the number of features selected, the r-sqaured score lowered significantly. This meant that the model would not have its best fitting without these predictor variables. On top of this, using min/max scaling as opposed to the StandardScaler library proved to not be as effective as the r-squared score would be significantly lower in that case as well.

A big issue we ran into with said linear regression model was that it would output negative values, so we had to change those too 0 since survival times cannot be negative. This model generally performed in the 50 to 59 percent range for the r-squared scores. This means that the model needs to be further developed to become an effective model. An assumption to fix this is to build the model with hyperparameters that are tuned to perform the best.

The Kaplan Meier is still a work in progress, so we do not have a result for it just yet.