Survival Analysis and Machine Learning Methodologies for Bone Marrow Cancer

Yitian Cai, Rudra Guin, Jorge Portugal

12/17/2021

Introduction

For this project, our group chose to perform statistical analysis using the Clinical Data Set from the Clinical Package of bone marrow cancer data. It was collected from the National Cancer Institute (NCI) genomic data portal. The version of the dataset we used was released by October 29, 2021. The bone marrow cancer can damage the bones, immune system, kidneys, and red blood cell count. The dataset contains 16,029 different cases of bone marrow cancer. We planned on using it for survival analysis and machine learning methodologies.

Data Description

The dataset contains 47 clinical variables. At the beginning, we tried to include the biospecimen data in order to increase the dimension of our dataset, but we found that most of those genomics data were missing, which would not provide us too much help. So, we decided to stick with the clinical data of bone marrow. Vital status means whether the patients were alive or dead at the time the data were collected. Our dataset contains 9,036 alive cases and 3,551 dead cases. Others were either missing or recorded as "unknown" or "not reported".

Data Preprocessing

The initial dataset had several missing values among the different features. So, before going into analysis and modeling, we had to clean up and pre-process the data. First, we removed all the observations that had missing values for vital status, because vital status was the one that we wish to forecast or predict and we cannot use missing values for prediction. We tried to see if the missing values in vital status were censored. However, none of the other variables that had similar information, such as days to last follow up, days to death, or days to birth, provided any useful information, since they were also missing. So, we decided to drop them all. Then, we decided to remove variables with more than 8,000 missing values, because it means that more than half of the observations in these variables were missing, which we believed would cause too much inaccuracy after data imputation. After that, we used the mice function to impute those missing values. The mice package implemented a method to deal with missing data and it generated Multivariate

Imputations by Chained Equations. And finally, we randomly split the dataset into two where the training set contained 70% of the data, and the test set contained the rest 30% of the data.

Analysis

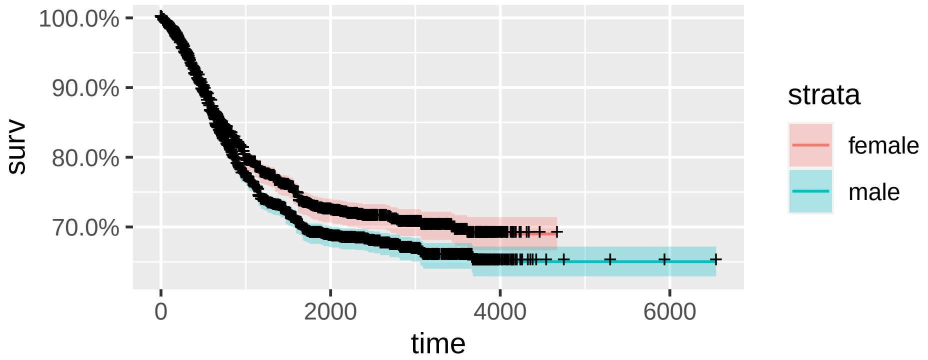
Survival Analysis

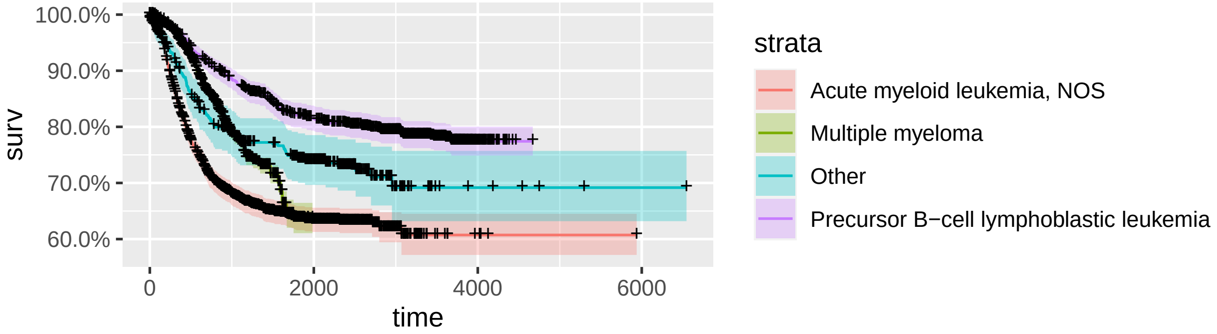
For the first part of this project, we decided to do some exploratory survival analysis. However, due to the nature of how the data was collected, as well as how vastly incomplete the data was, some assumptions had to have been made. After looking at the dataset dictionary, the definition of the variable Vital Status is defined as “the survival state of the person registered on the protocol”. That is the only clear definition given out for that variable. As a result, we make the assumption that even though a patient’s info is censored, we consider them dead.

Additionally, our time event variable is “days to last follow-up” . We were recommended to go over the “days to death” variable and perhaps use that as our time event variable, but upon further examination, that variable had a large proportion of missing values and highly skewed the data.

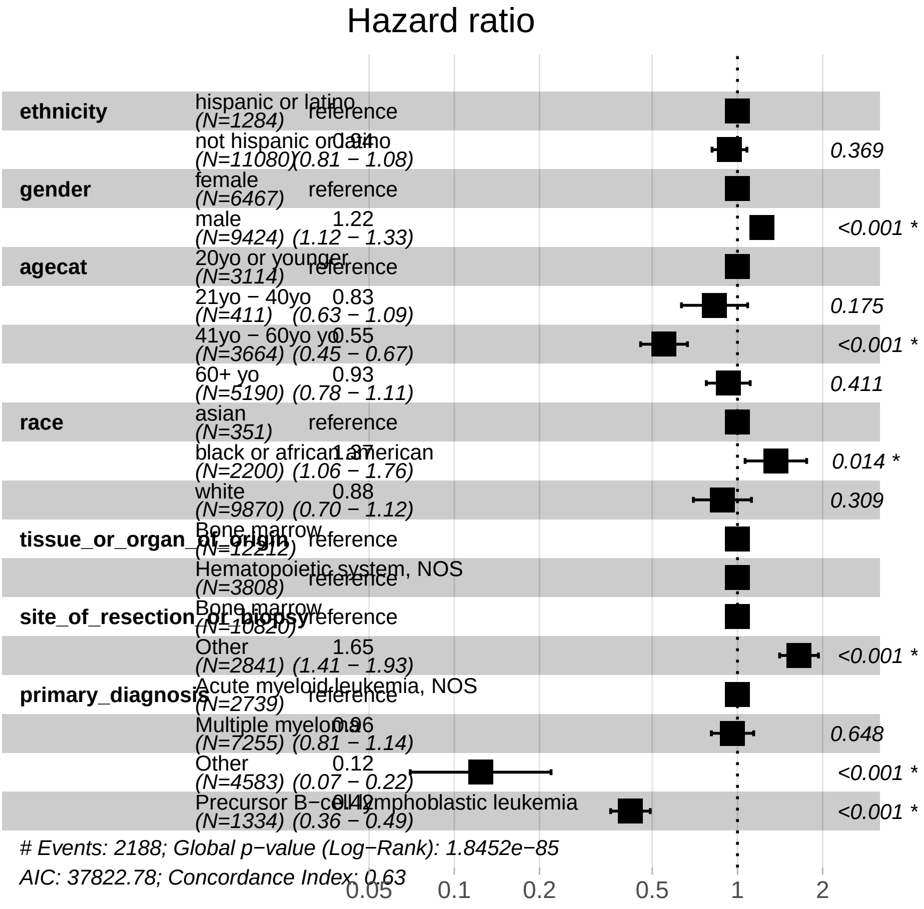
Our main goal was to obtain results from machine learning methods. While the Survival Analysis portion may not follow the traditional guidelines (due to the vastly incompleteness of the dataset). It is mostly meant to be an exploratory analysis to give us and the readers a feel for the data at hand.

Initially, Kaplan-Meier plots were fit on the data for visualization purposes, the first plot other than the average survival rate for everyone drops to about 77% after 1000 days, but the dropoff levels off afterwards. (Plot was ommited due to space constraints.For better visualization, KM plots were generated for gender and for primary disease diagnosis. From here, we can see that females have a higher survival rate than males; and that Precursor B-cell lymphoblastic leukemia was disease with the highest survival rate, whole Acute myeloid leukemia had the lowest survival rate for the entire time period.





Some of the more complete variables that had a relation to Vital Status were: age, gender, ethnicity, race, tissue or organ of origin, site of resection or biopsy, and primary disease diagnosis. Using these variables, A Cox proportional Hazards model was constructed to see the effect these variables had on Vital Status, as well as to check if said effects were significant. A forest plot was used to depict these results. From this forest plot, we can see that the most significant factors were: if the individual was a male (p-value <.001), if they were in the 41-60 age range (p-value <.001), if they were black/African-American (p-value =.014), if the site of resection or autopsy was anywhere other than bone marrow (p-value <.001), or if their primary disease diagnosis was Precursor B-cell lymphoblastic leukemia (p-value <.001).

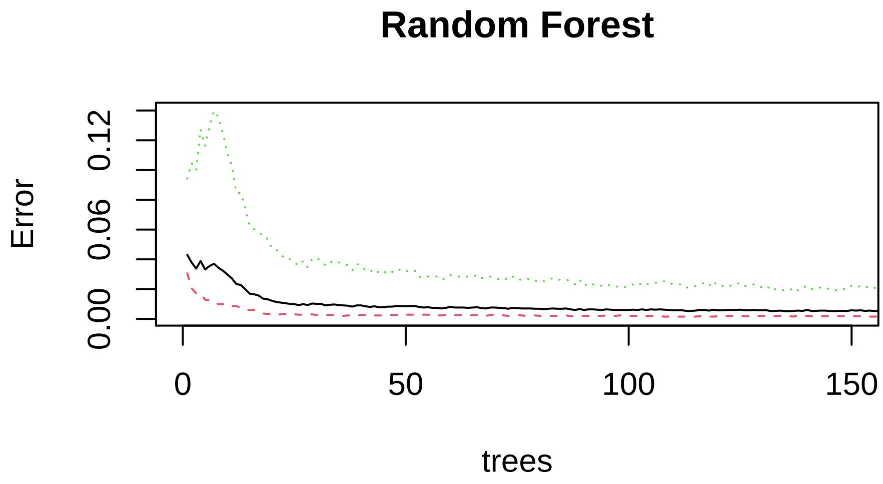


Machine Learning Modeling and Analysis

We also performed some machine learning modeling and evaluated the effectiveness of the models' prediction power of the vital status of certain patients.

Models

Random Forest After the missing values were imputed, we first tried to apply the random forest algorithm to that dataset. In order to fit the dataset into the model, we needed to remove categorical variables that had only one level since they were not suitable for random forest models. Then we ran the random forest algorithm using the training set that contains 70 percent of the original data, and we got the following plot.



The black line in the plot showed the out-of-bag (OOB) error that measuring the prediction error of random forests. We could see that when the number of trees is about 20, the prediction error was kind of low and the error kept nearly unchanged when the number of trees increased. So, we decided to choose 20 as the number of trees for our random forest algorithm. The following table showed the overall accuracy of the prediction using the random forest model.

Neural Network After making dummy variables for every categorical feature and normalizing each numerical feature, several instances with missing values were removed, therefore the total number of instances in the overall neural network dataset was 6,916, before it was split into training and testing subsets. Prior to the removal of instances with missing values for the numerical features, there were a total of 12,587 instances in the dataset.

Model Results

```
## [1] "Model accuracies across training and testing sets:"
```

Model	Training.Testing.Set	Accuracy
Random Forest	Training	0.9997936
Random Forest	Testing	0.9942057
Neural Network	Training	0.9873993
Neural Network	Testing	0.9739759

```
## [1] "Random forest testing set confusion matrix:"
```

```
##           Reference
## Prediction Alive Dead
##      Alive  1656    8
##      Dead      4  403
```

```
## [1] "Neural network testing set confusion matrix:"
```

```
##           Reference
## Prediction    0    1
##          0  377   33
##          1   21 1644
```

Results and Conclusion

For the random forest algorithm with 20 trees, the accuracy of prediction for both the training set and the testing set were more than 99.5%. The accuracy would still be above 99% if we changed the number of trees to 10. The performance of the prediction looked great, but such a high accuracy in both the training set and testing set may indicate some potential inflexibility when applying the same model to a totally new dataset. For the neural network model, with 2 hidden layers consisting of 50 nodes first and then 10 nodes, the accuracy of prediction for both the training set and the testing set were more than 97%. The results of this model is quite promising, as the neural network was able to make many accurate predictions, but they were slightly less than the results of the random forest model. Same as what mentioned in the previous model, such a high accuracy in both the training set and testing set may indicate problems when the same algorithm is applied to a completely different dataset. Hence, it may be ideal to re-train both models when we work on a brand new bone marrow cancer data set.

Additionally, it takes under 50 minutes to generate the neural network model, trained using the training subset. It could probably be sped up using cloud computing services such as Great Lakes, AWS, etc.