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 mice() to impute those missing values. The mice package implemented a method to deal with missing data and it generates 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

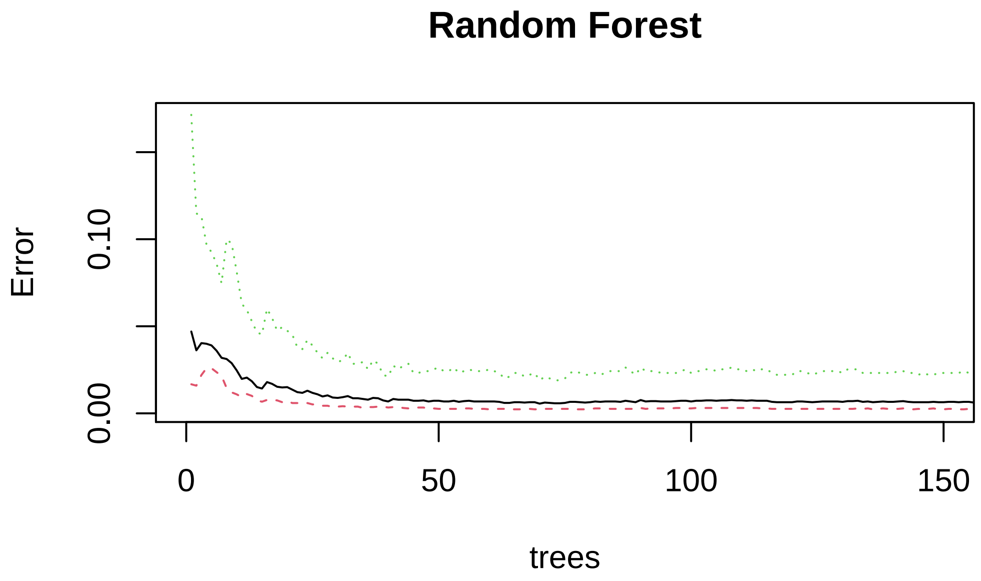
We first conducted some survival analysis

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.

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.

```
## [1] "Training set accuracy:"

##  Accuracy
## 0.9995872

## [1] "Testing set confusion matrix and accuracy:"

##           Reference
## Prediction Alive Dead
##      Alive  1656    1
##      Dead      4  410

##  Accuracy
## 0.9975857
```

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.

```
## [1] "Training set accuracy:"
```

```
##  Accuracy
## 0.9995872
```

```
## [1] "Testing set confusion matrix and accuracy:"
```

```
##           Reference
## Prediction Alive Dead
##      Alive  1656    3
##      Dead      4  408
```

```
## Accuracy
##  0.99662
```

Results and Conclusion

For the random forest algorithm with 20 trees, the accuracy of prediction for both the training set and testing set were more than 99.5%. The accuracy would still be above 99% if we changed the number of trees to 10. The perforamce of the prediction looked great, but such a high accuracy in both training set and testing set may indicate some potential problems when applying the same algorithm to a totally new dataset. So, we have to be cautious and re-train our model when we encounter a new set of bone marrow cancer data.

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, etc.

Division of Work:

Yitian: worked on initial data preprocessing by transforming all variables into numeric or factor, then worked on data imputation and data cleaning, and finally implemented and evaluated the random forest model.

Rudra: worked on initial data preprocessing by converting empty values to `NA`, then did further preprocessing of the data for the neural network model, and finally implemented and evaluated the neural network model.

Jorge: