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 has several missing values among the different features. So, before going into analysis and modeling, we first clean up and pre-process the data. First, we remove all the observations that have missing values for vital status, because vital status is the one we wish to forecast or predict. 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 believe will cause too much inaccuracy after data imputation. After that, we used mice() to impute those missing values. The mice package implements 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 contains 70% of the data, and the test set contains the rest 30% of the data.

Analysis

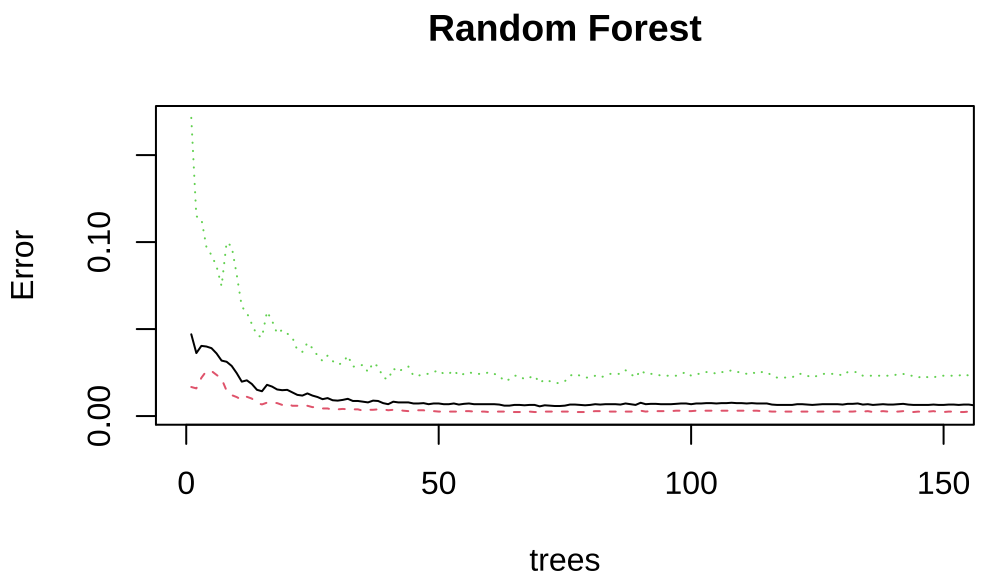
Survival Analysis

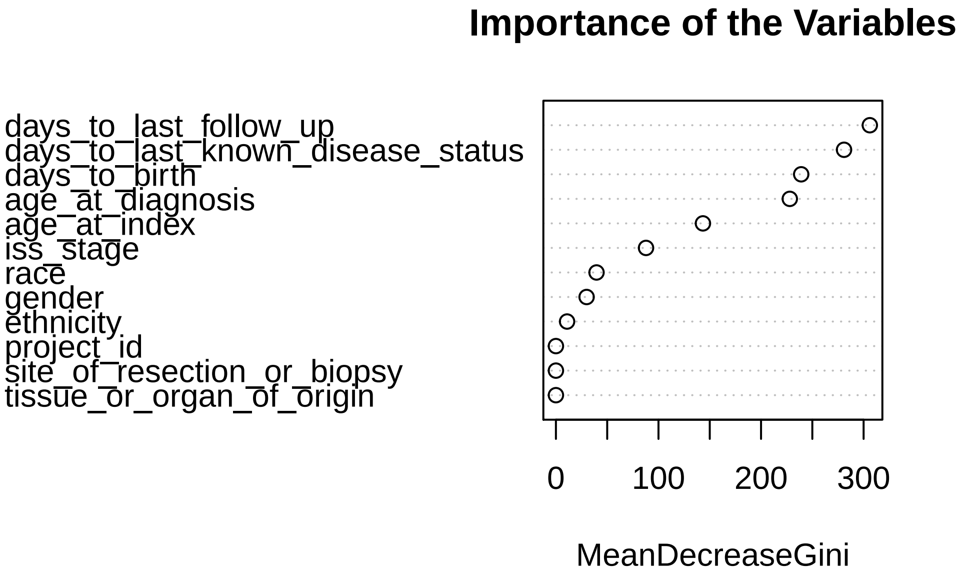
We first conducted some survival analysis

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





```
# Results:

p1 <- predict(rf2, train_rf)

confusionMatrix(p1, train_rf$vital_status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Alive Dead
##      Alive  3896    2
##      Dead      0  947
##
##                Accuracy : 0.9996
##                  95% CI : (0.9985, 1)
##     No Information Rate : 0.8041
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9987
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 1.0000
##             Specificity : 0.9979
##          Pos Pred Value : 0.9995
##          Neg Pred Value : 1.0000
##              Prevalence : 0.8041
##          Detection Rate : 0.8041
##    Detection Prevalence : 0.8045
##       Balanced Accuracy : 0.9989
##
##        'Positive' Class : Alive
##
```

```r
p2 <- predict(rf2, test_rf)
confusionMatrix(p2, test_rf$vital_status)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction Alive Dead
##      Alive  1656    1
##      Dead      4  410
##
##                Accuracy : 0.9976
##                  95% CI : (0.9944, 0.9992)
##     No Information Rate : 0.8015
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9924
##
##  Mcnemar's Test P-Value : 0.3711
##
##             Sensitivity : 0.9976
##             Specificity : 0.9976
##          Pos Pred Value : 0.9994
##          Neg Pred Value : 0.9903
##              Prevalence : 0.8015
##          Detection Rate : 0.7996
##    Detection Prevalence : 0.8001
##       Balanced Accuracy : 0.9976
```

```
##
##        'Positive' Class : Alive
##
```

Neural Network

```
## Time difference of 5.418807 mins
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  940  398
##          1 1501 5971
##
##                Accuracy : 0.7844
##                  95% CI : (0.7757, 0.793)
##     No Information Rate : 0.7229
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3748
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3851
##             Specificity : 0.9375
##          Pos Pred Value : 0.7025
##          Neg Pred Value : 0.7991
##                Prevalence : 0.2771
##            Detection Rate : 0.1067
##      Detection Prevalence : 0.1519
##         Balanced Accuracy : 0.6613
##
##          'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  407  184
##          1  703 2483
##
##                Accuracy : 0.7652
##                  95% CI : (0.7513, 0.7786)
##     No Information Rate : 0.7061
##     P-Value [Acc > NIR] : 2.32e-16
##
##                   Kappa : 0.3447
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3667
##             Specificity : 0.9310
##          Pos Pred Value : 0.6887
##          Neg Pred Value : 0.7793
##              Prevalence : 0.2939
##          Detection Rate : 0.1078
##    Detection Prevalence : 0.1565
##       Balanced Accuracy : 0.6488
##
##        'Positive' Class : 0
##
```

Results

Conclusion

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: