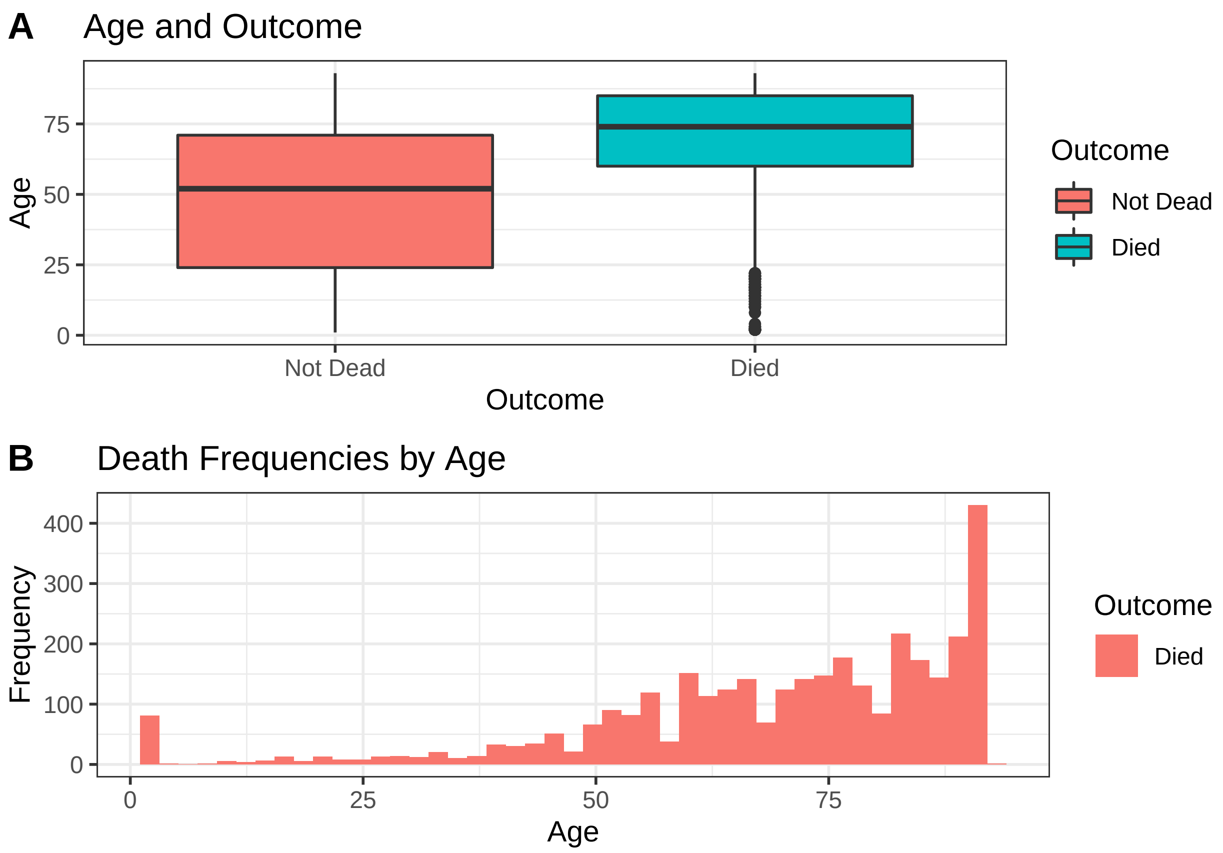
PM591 - Final Project

Brandyn Ruiz

4/6/2021

Introduction

Death has always been a major fear for many and is the natural course of life. In our society we have always associated death with hospitals as it shown in popular movies like Disney's Up. With ongoing advancements in medical science and technology a person's life expectancy has risen compared to centuries ago. However, there are the unfortunate cases of deaths within a hospital stay and all these records have been recorded by the National Impatient Sample (NIS) and its data collected by the Healthcare Cost and Utilization Project (HCUP). Hospitals themselves should not be looked on as the causation of death when a patient is admitted, as the NIS has a wealth of information of each inbound patient with records of their drug use `ARPDRG_Severity`, length of stay (`LOS`), `RACE`, `AGE`, `HODP_DIVISION`, if the patient was transferred in (`TRAN_IN`), and operating procedure to name a few. With this vast dataset statisticians and hospitals can use these variables of interests to predict death for another inbound patient, to gauge a metric of how likely the patient is to die during their hospital stay with their incoming conditions and their personal demographics of age, race, and gender. Predicting this particular outcome of death will help to improve the efficiency of hospitals and to better improve the quality of life for those admitted into the hospital, as nurses and doctors will be able to provide the proper attention and care for those who are likely to die from their current conditions in the prediction model. The (NIS data) [https://www.hcup-us.ahrq.gov/] can be found using the link and this project will be looking into a subsample of 200,000 of all hospital patients in 2012. Generally looking at the death outcome across the ages of the patients, we see the total distribution in the box and whisker plot with death occurrences at the minimum age, the three quartile ranges, and the maximum values with plotted outliers. The occurrence of death is common for patients that are much older in age towards 60 years and above, but we still have occurrences amongst those that are relatively young although not very many and are marked as outliers. In the next plot we see the frequency of death occurrence amongst all the ages and can see the quadratic trend as age increases so does death occurrences. Which is natural in a hospital environment as sickness and symptoms affect those elderly and with much weaker immune systems. However, we also see the spike for those prenatal and infants as there can be numerous complications with childbirth.



Methods

The sampled NIS dataset is a real-world dataset that is not preprocessed and there are a few missing observations that would conflict with our prediction models such as random forests and boosting models. There are a total of 13,904 observations missing and some with our main outcome of interest `Death`, we delete these missing observations and are left with 186,096 observations for our prediction models. Out dataset has 157 variables recorded for each patient and running our prediction model for a patient's death we use selected variables of what the hospitals would have records of at the time of admission. The primary interest of our prediction model is to predict the death of an admitted patient and some variables may be recorded during their stay at the hospital and those observations would not help our model to predict death for an entirely new patient with different records before reaching the hospital. Therefore, we selected variables such as their demographics like `age`, `gender`, `race`, and what month they were admitted `amonth`, if it was the weekend or not `aweekend`, their drug severity `APRDRG_Severity`, their current health conditions if they have `CM_AIDS`, are alcholics `CM_Alcohol`, have churned lung `CM_CHRNLUNG`, are obese `CM_OBESE`, if they elected to go to the hospital themselves `ELECTIVE`, if they had an operation procedure `ORPROC`, their method paying their hospital bill `PAY1`, and if they were transferred in from a different medical facility `TRAN_IN` to name a short few.

I have used three different types of predictive models each with progressive performance. The first I have used was elementary logistic regression learned from basic statistics classes with a categorical outcome of whether the patient died or not and fitting all our parameters of interests with our main outcome of death to find the log likelihood of death occurring. We fit all the parameters of interest because these are readily available for each inbound patient and will give our model predictive accuracy. Then using machine learning techniques such as random forests to decorrelate trees that have positive correlated samples, to reduce variance. Since our prediction model is a classification problem of those that died and did not we consider a random subset of $\sqrt(46)$ of all our possible variables of interest. Our final prediction model is boosting as a general-purpose algorithm to improve the prediction of base learners like decision trees. Boosting sequentially improves the prediction by fitting a model to the residuals, updating the model by adding to the residuals and continues this process until the limit of iterations is met or the prediction model reaches the smallest cross-validation error.

Using machine learning techniques to split our subset of the NIS data with our selected features, we use 70% for the training set (N = 130266) and 30% for the testing set (N = 55830). This gives us a substantial amount to train our prediction models on the training set. Then to make our test prediction of the predictive model on the test set. Which we then compare the performance of our three predictive model's sensitivity, specificity, accuracy, and misclassification error rate. We use accuracy as a performance metric to describe how often our model is accurate in classifying a death outcome correctly, Accuracy $= \frac{\text{(True Positive + True Negative)}}{\text{Total}}$. Misclassification error rate is the opposite of accuracy and describes how often the model is wrong in classifying our outcome of interest death, Misclassification $= \frac{\text{(False Positive + False Negative)}}{\text{Total}}$. In predicting deaths for a patient, we would want the most accurate model in predicting death with the lowest misclassification error rate, this information would help the hospital and could have the potential to save a life. Sensitivity is another performance metric when our actual outcome is death (the positive case), how often does it actually predict the death and is given by the formula Sensitivity $= \frac{\text{True Positive}}{\text{Actual Yes}}$. Specificity tells us when the reality is not a death, and how often does the model predict a negative case (did not die), and is given by Specificity $= \frac{\text{True Negative}}{\text{Actual No}}$.

Results

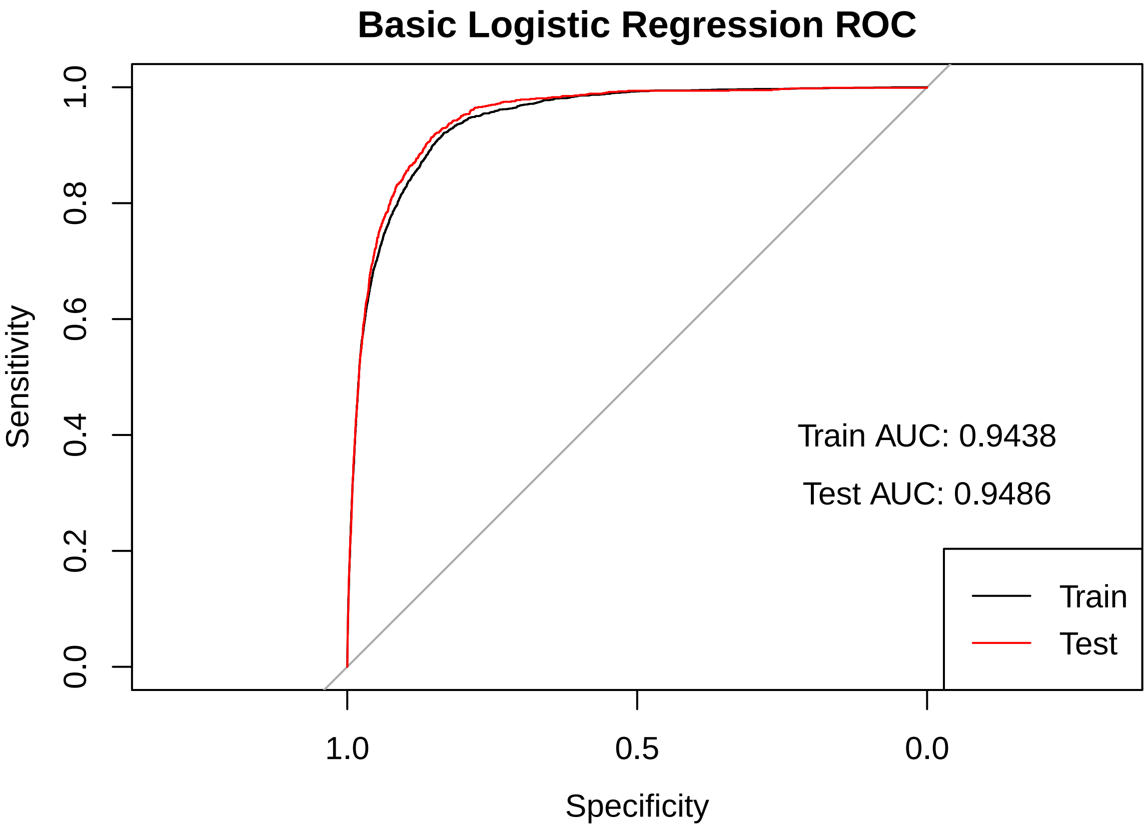
In running our basic logistic regression to classify the positive case of a death occurrence we achieve a training accuracy of 0.9820, which performs well in classifying a death outcome. However, our base model has a sensitivity rate of 0.0770 and a specificity rate of 0.9988, which can be referenced on the table output below.

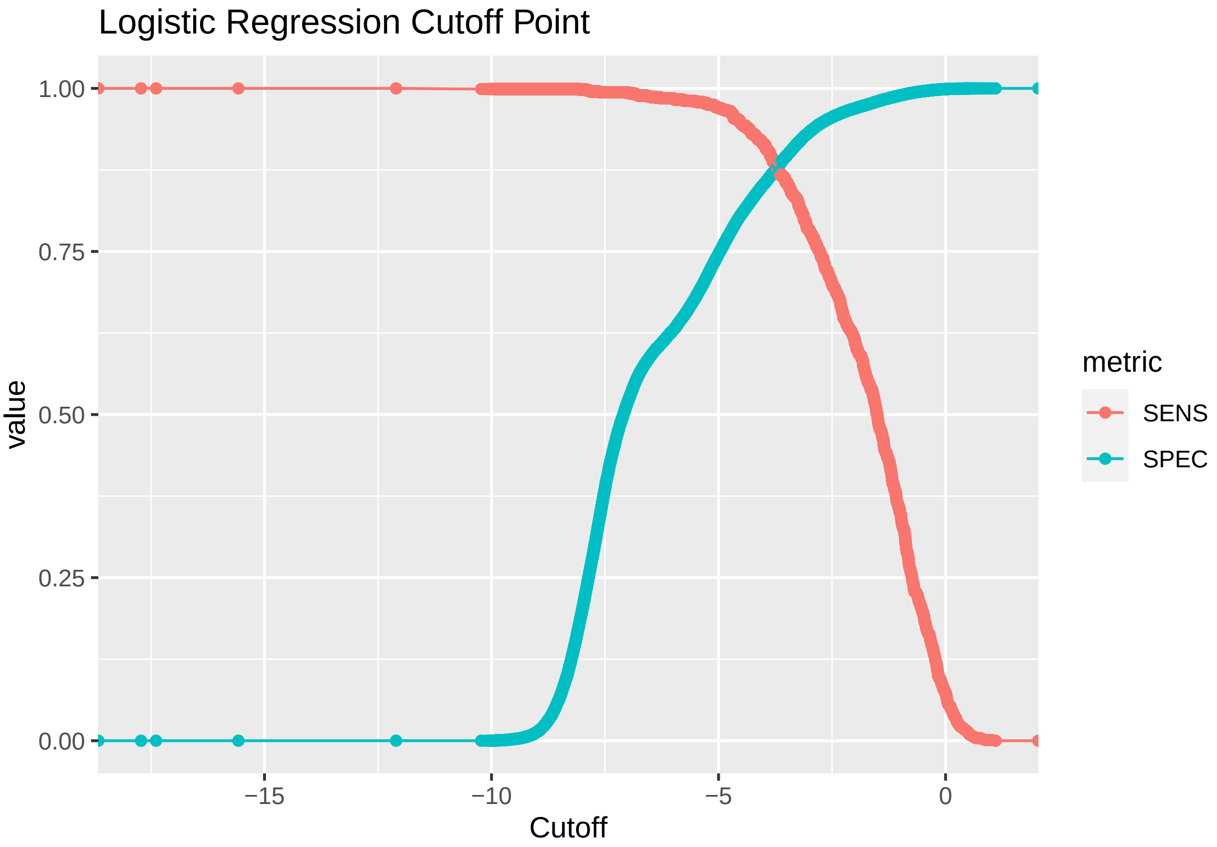
Basic Logisitic Regression

```
##
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   Died Not Dead
##   Died        183      148
##   Not Dead   2193   127743
##
##                Total n : 130'267
##               Accuracy : 0.9820
##                 95% CI : (0.9813, 0.9827)
##    No Information Rate : 0.9818
##    P-Value [Acc > NIR] : 0.2380
##
##                  Kappa : 0.1313
##  Mcnemar's Test P-Value : < 2.2e-16
##
##            Sensitivity : 0.0770
##            Specificity : 0.9988
##         Pos Pred Value : 0.5529
##         Neg Pred Value : 0.9831
##             Prevalence : 0.0182
##         Detection Rate : 0.0025
##   Detection Prevalence : 0.0014
##      Balanced Accuracy : 0.5379
##         F-val Accuracy : 0.1352
##      Matthews Cor.-Coef : 0.2016
##
```

```
##        'Positive' Class : Died
```

With such a low sensitivity rate this does not help for our predictive model to correctly identify a patient that would die. I ran a cutoff point analysis that would indicate which optimal cutoff point would give us the greatest sensitivity rate and specificity rate, as each model would have a tradeoff between both rates. Referencing from the cutoff plot, we are given the entire distribution of sensitivity rates and specificity rates, the optimal cutoff point would be where the two measures of performance would intersect and would be -3.9749, giving us a sensitivity rate of 0.9136 and a specificity rate of 0.8553. Our training and test datasets with their respective `AUC` values of 0.9438 and 0.9486 gives us great model predictive performance.





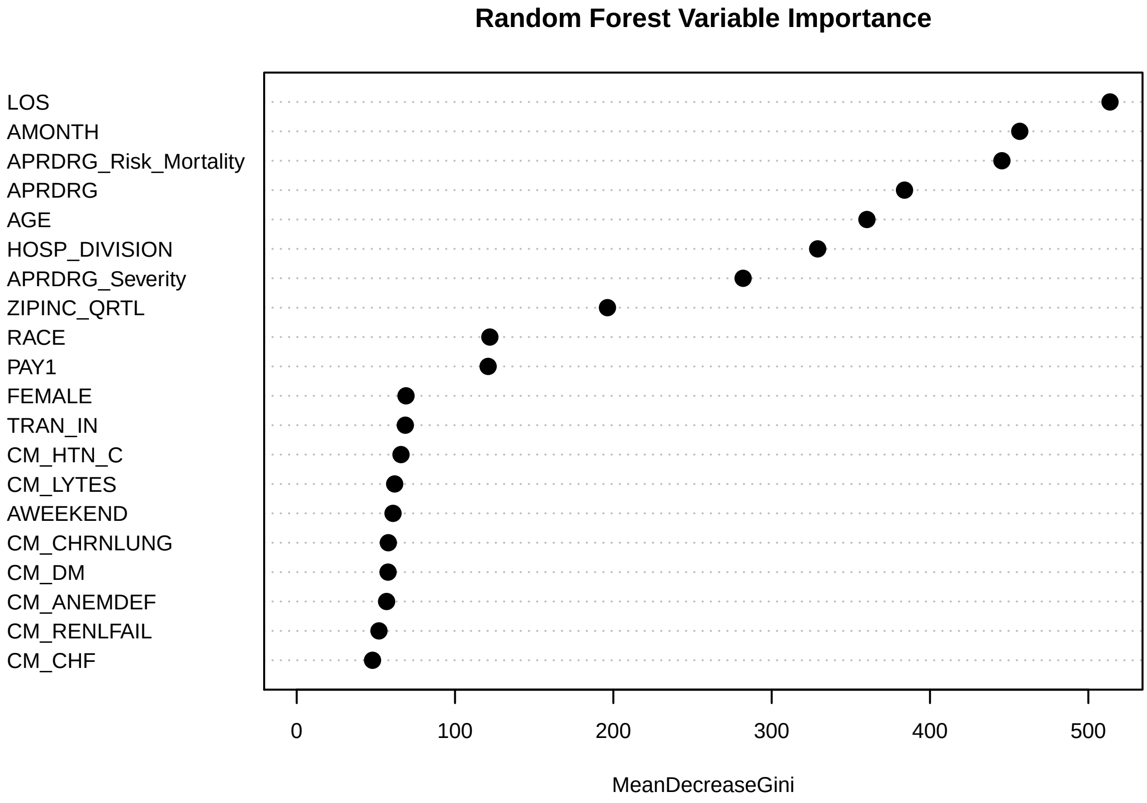
```
## # A tibble: 10 x 4
##    Cutoff  SENS  SPEC   SUM
##     <dbl> <dbl> <dbl> <dbl>
##  1  -3.97 0.914 0.855  1.77
##  2  -3.98 0.914 0.855  1.77
##  3  -3.98 0.914 0.855  1.77
##  4  -3.98 0.914 0.855  1.77
##  5  -3.98 0.914 0.855  1.77
##  6  -3.98 0.914 0.855  1.77
##  7  -3.98 0.914 0.855  1.77
##  8  -3.98 0.914 0.855  1.77
##  9  -3.98 0.914 0.855  1.77
## 10  -3.98 0.914 0.855  1.77
```

Further improving our predictive model, we use more advanced models in machine learning such as random forests. After splitting our subsampled data of the NIS for our selected features of interest, we have a training set with 127907 patients classified as `Not Dead` and 2359 have `Died`. In using the `randomForest` package, our confusion matrix for the Random Forest model gives us a misclassification error for those that did not die compared to those that did die, a value of 0.0006 and 0.8711, respectively. Since the Random Forest model is classifying to the majority being those that did not die, our outcome of interest those that did die is relatively low.

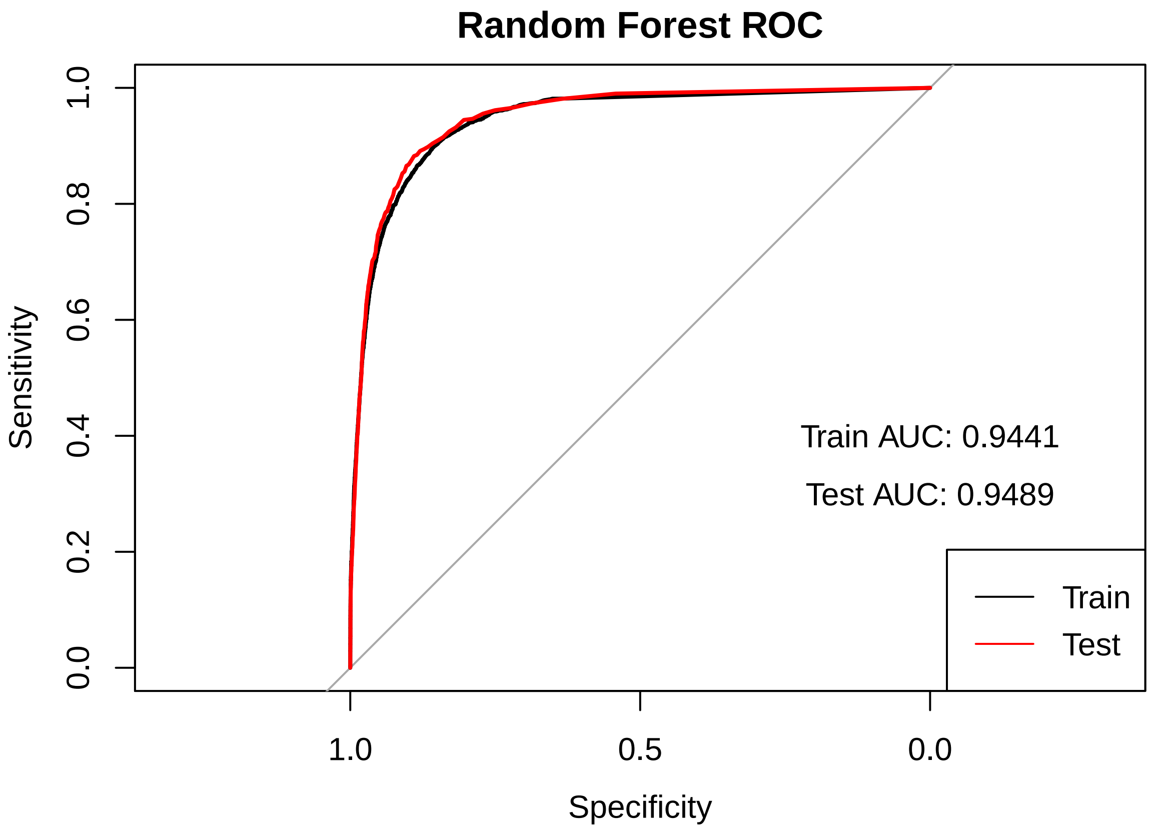
Random Forest

```
##          Not Dead Died  class.error
## Not Dead   127827   80 0.0006254544
## Died         2055  304 0.8711318355
```

Our Random Forest variable importance output referenced below lists length of stay (`LOS`), admission month (`AMONTH`), their risk of mortality (`APRDRG_Risk_Mortality`), drug usage (`APRDRG`), and `AGE` as the top five features with the most importance to our model. The Receiver Operating Characteristic (ROC) curve output shows the entire performance of our model's sensitivity and specificity rates. Even with good predictive performance of our training and test datasets with `AUC` values of 0.9441 and 0.9489 respectively, which performs better than our basic logistic regression model. Our Random Forest model would not be the ideal predictive model for deaths within hospital as it overclassifies to the majority class being our negative case of those that did not die.



```
##        Not Dead  Died
## 12598     0.996 0.004
## 145339    1.000 0.000
## 163832    1.000 0.000
## 121796    0.998 0.002
```

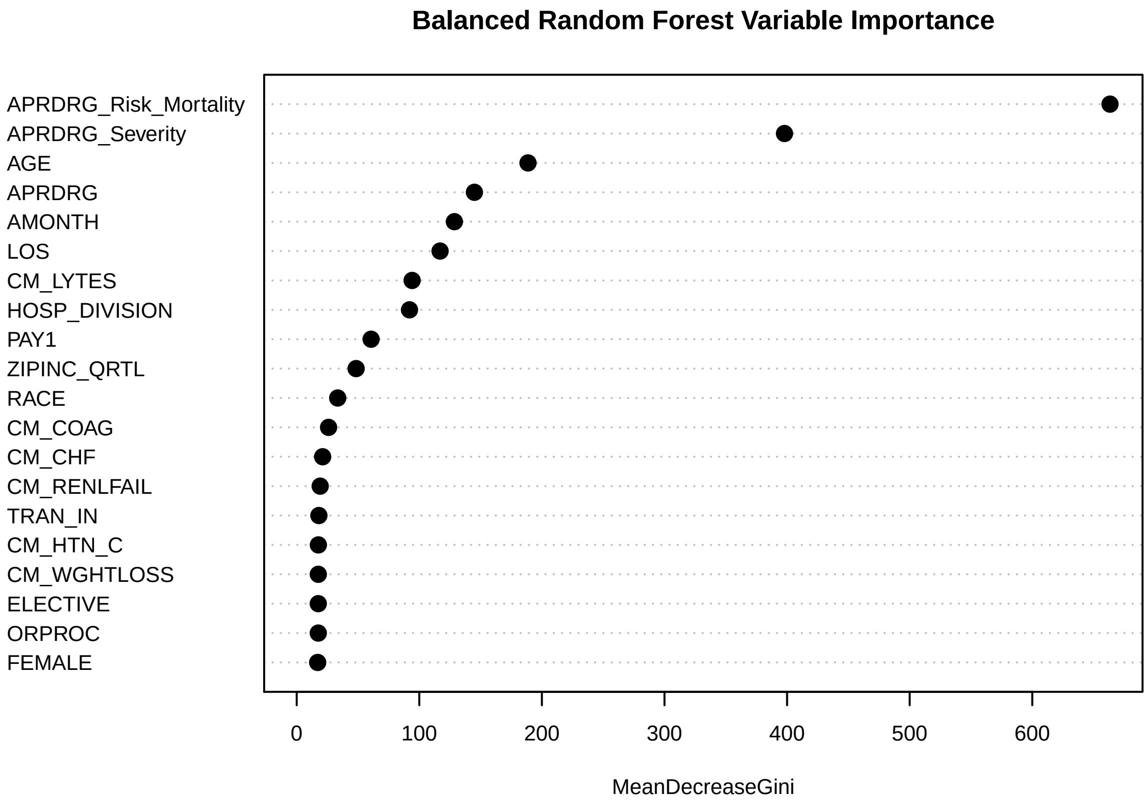


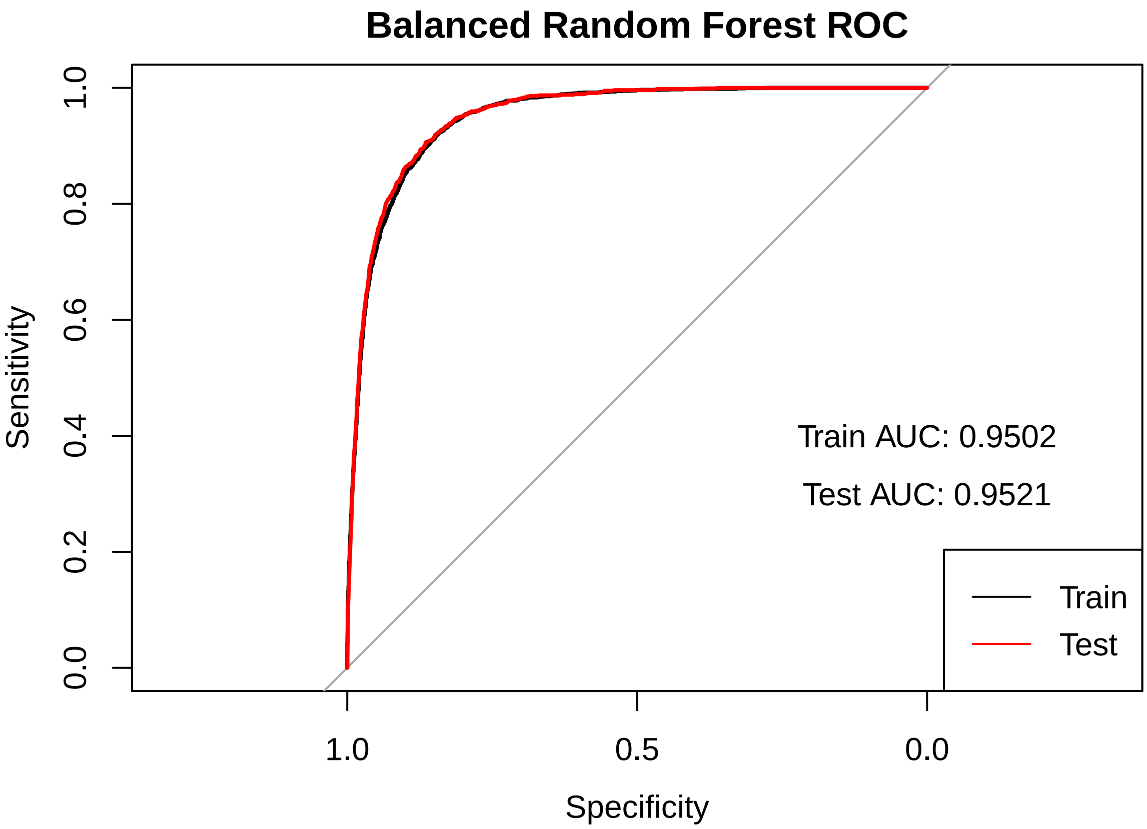
We then alternatively turn to a Balanced Random Forest prediction model, using the same training and test sets but with sampling the negative case to match the same sample size as the positive cases of the death outcome. We see a significant improvement in the misclassification error rates from the confusion matrix output below, that those who are not dead have a misclassification error of 0.1193 and for those that have died with a value of 0.1242.

Balanced Random Forest

```
##          Not Dead  Died class.error
## Not Dead   112647 15260   0.1193054
## Died          293  2066   0.1242052
```

In balancing the majority class to the same sample size as our outcome of interest our Balanced Random Forest variable importance changes with the top five variables being `ARPDRG_Risk_Mortality`, `ARPDRG_Severity`, `AGE`, `APRDRG`, and `AMONTH`. However, for our Balanced Random Forest model the ROC curve output has a training and testing `AUC` values of 0.9502 and 0.9521 respectively which is another significant improvement of predictive model performance compared to the basic logistic regression and unbalanced Random Forest models.


Balanced Random Forest Variable Importance
APRDRG_Risk_Mortality
APRDRG_Severity
AGE
APRDRG
AMONTH
LOS
CM_LYTES
HOSP_DIVISION
PAY1
ZIPINC_QRTL
RACE
CM_COAG
CM_CHF
CM_RENLFAIL
TRAN_IN
CM_HTN_C
CM_WGHTLOSS
ELECTIVE
ORPROC
FEMALE
0
100
200
300
400
500
600
MeanDecreaseGini

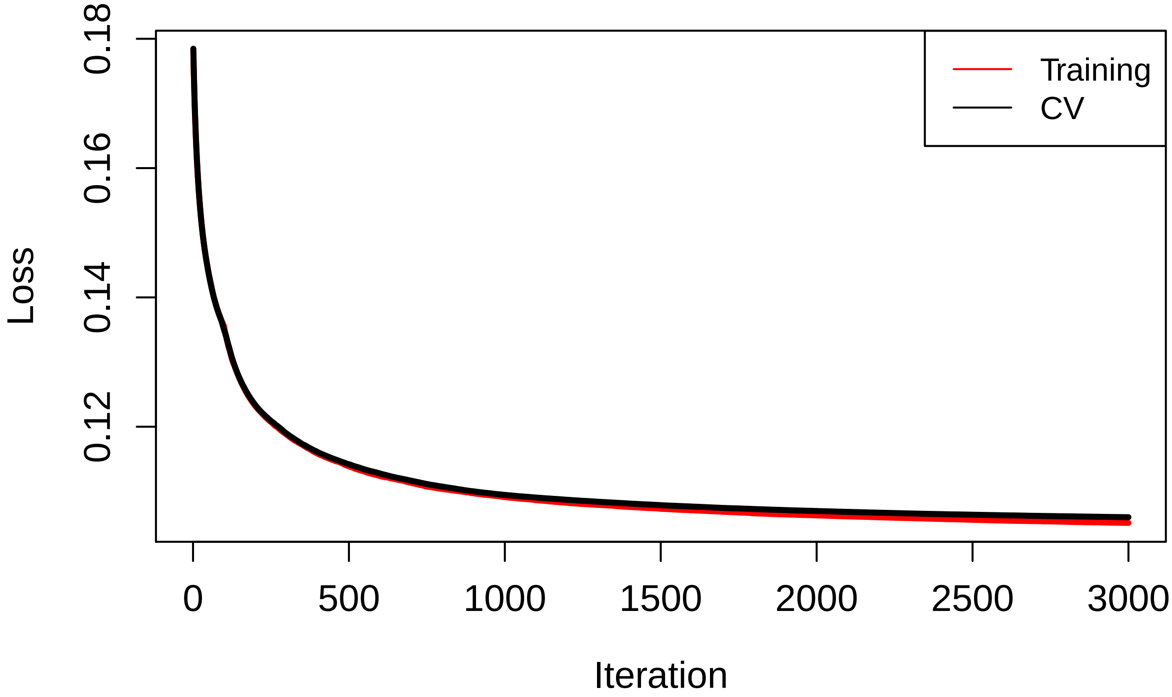


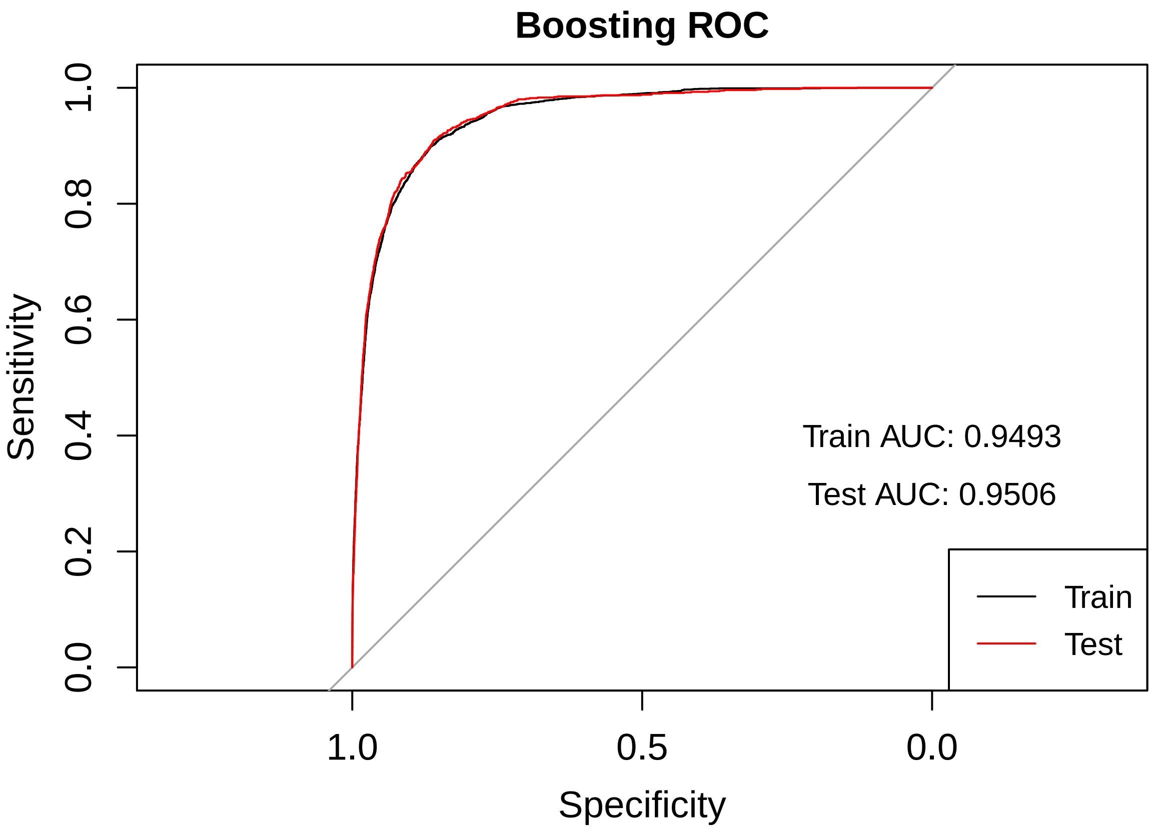
Lastly, we try another sophisticated machine learning predictive model with Boosting, with model parameters of a shrinkage rate that learns faster, but at a greedily pace with $\lambda = 0.01$ with interaction depth 1 and 10-fold cross-validation methods. In predicting such a lengthy dataset, a greater λ value would be less computationally expensive. Our optimal number of trees is at a value of 3,000 which we use for our predictions to generate the ROC curves. The cross-validation plot presents the training error and cross validation error against `LOSS`, the logistic loss function instead of using residual sum of squares as the metric of measure. In our Boosting predictive model, we have another change in variable importance with the top five being `ARPDRG_Risk_Mortality`, `ARPDRG_Severity`, `LOS`, `ARPDRG`, and `AGE`. The ROC curve for our Boosting prediction model has training and test `AUC` values of 0.9493 and 0.9507, respectively.

Boosting

```
## gbm(formula = 1 * (DIED == "Died") ~ ., distribution = "bernoulli",
##     data = NIS_simple[train, ], n.trees = 3000, interaction.depth = 1,
##     shrinkage = 0.01, cv.folds = 10, class.stratify.cv = TRUE)
## A gradient boosted model with bernoulli loss function.
## 3000 iterations were performed.
## The best cross-validation iteration was 3000.
## There were 45 predictors of which 33 had non-zero influence.
```

```
##                                         var    rel.inf
## APRDRG_Risk_Mortality APRDRG_Risk_Mortality 71.4890050
## APRDRG_Severity             APRDRG_Severity 15.4253524
## LOS                                     LOS  4.8232066
## APRDRG                               APRDRG  3.1952667
## AGE                                     AGE  2.1905463
## CM_METS                             CM_METS  0.8082817
## CM_COAG                                 CM_COAG  0.6597185
## CM_ANEMDEF                           CM_ANEMDEF  0.2216164
## CM_LIVER                               CM_LIVER  0.1932715
## CM_PARA                                 CM_PARA  0.1533456
```





Discussion

Our prediction models have achieved high sufficient accuracy with training and test AUC values that are close to one another and agree with each other. I would not use the unbalanced Random Forest as our prediction model to predict impatient mortality as we have seen that deaths within the hospital are not as common and are not the majority class. With selecting our available features for a patient's admission to a hospital I was surprised to see that variables such as `ORPROC` whether a patient had a major operating room procedure, `HOSP_DIVISION` the census division of the hospital divided into 9 groups amongst the United States, and `TRAN_IN` whether a patient was transferred from a different acute care hospital, or another type of health facility, or even not transferred at all did not show as often or have a high variable importance in our prediction models. Before making our models, I preselected these particular features because of how influential they could potentially be for a patient and their inbound mortality within a hospital stay. For example, the categorical variable of major operating procedure `ORPROC`, if a patient were to have open heart surgery, brain surgery, or even a heart transplant depending on the current condition and severity of the patient they would be more likely to die during the operation. Even with `HOSP_DIVISION` as some regions within the States could have hospitals not as accessible to all in different parts as hospitals are located in the urban and suburban environments compared to rural and patients may have the difficulty in reaching the hospital for their conditions with adequate time and even hospitals in divisions where snow and weather conditions could prohibit the patient's admission to reaching and getting inside the hospital to receive the proper care in time. `TRAN_IN` if a patient were to be transferred to a better equipped hospital for a certain procedure or a medical doctor's lifesaving expertise, for a patient to be even transferred would mean their condition is in severity and the transfer could have had an impact of their likelihood to die within admission. Further considerations for better prediction models would be to consider interaction effects as we may see a significant effect of `OPRPROC` and `TRAN_IN` as a combined effect of the potential of mortality for a patient to be transferred for a major operating procedure.