Data Challenge: AdaBoost Model Fitting

Aldo Iturrios

3/28/2021

Training Data

```
# Data
train_data.sumvars <- read.csv("../Data/clean_data/train_data_sumvars.csv")
test_data.sumvars <- read.csv("../Data/clean_data/test_data_sumvars.csv")
test_outcomes <- read.csv("../Data/outcomes/test_nolabel.csv")

# Remove Mean, Min, and Max Summary Variables (Not Important to Model)
var_remove <- grep("_mean", colnames(train_data.sumvars))
train_data.sumvars <- train_data.sumvars[, -var_remove]
var_remove <- grep("_min", colnames(train_data.sumvars))
train_data.sumvars <- train_data.sumvars[, -var_remove]
var_remove <- grep("_max", colnames(train_data.sumvars))
train_data.sumvars <- train_data.sumvars[, -var_remove]
```

Training Validation Split (70% vs. 30%)

```
set.seed(08212021)
n = dim(train_data.sumvars)[1]
train_id = sample(seq(1, n, 1), floor(n*0.7))

# Validation Training Set (70%)
val.train = train_data.sumvars[train_id, ]

# Validation Test Set (30%)
val.test = train_data.sumvars[-train_id, ]
```

Paramater Tuning

```
# Hyperparameter grid
hyper_grid <- expand.grid(
  cutoff = c(0.6, 0.65, 0.7),
  n.trees = c(500, 1000),
  shrinkage = c(.01, 0.05),
```

```
  interaction.depth = c(3, 4, 5)
)

# Total number of combinations
nrow(hyper_grid)
```

```
## [1] 36
```

Function that does crossvalidation on all combinations of paramaters (listed above)

```
Kfold_CV_adaboost <- function(K, param_grid, param_combo, train) {

  fold_size = floor(nrow(train)/K)

  cv_error = rep(0,K)
  auc_score = rep(0,K)

  for(i in 1:K) {

    # iteratively select K-1 folds as training data in CV procedure, remaining as test data.
    if(i!=K){
      CV_test_id = ((i-1)*fold_size+1):(i*fold_size)
    }else{
      CV_test_id = ((i-1)*fold_size+1):nrow(train)
    }

    CV_train = train[-CV_test_id,]
    CV_test = train[CV_test_id,]

    # Fit logistic regression model
    ada_model <- gbm(outcome ~.,
                     data = CV_train,
                     distribution = "adaboost",
                     n.trees = param_grid$n.trees[param_combo],
                     interaction.depth = param_grid$interaction.depth[param_combo],
                     shrinkage = param_grid$shrinkage[param_combo])

    # Predict
    pred = predict(ada_model, newdata = CV_test, n.trees = param_grid$n.trees[param_combo], type = "resp

    # Predicted classifications
    ada_pred <- ifelse(pred > param_grid$cutoff[param_combo], 1, 0)
    pos_error <- mean(ada_pred[which(CV_test$outcome == 1)] != CV_test[CV_test$outcome == 1,]$outcome)
    neg_error <- mean(ada_pred[which(CV_test$outcome == 0)] != CV_test[CV_test$outcome == 0, ]$outcome)

    # Calculate CV error by taking averages
    cv_error[i] = (pos_error + neg_error) / 2

    # AUC Score
    pr <- prediction(pred, CV_test$outcome)
    auc = performance(pr, "auc")
```

```
    auc_score[i] <- as.numeric(auc@y.values)

  }
  return(c(mean(cv_error), min(cv_error), max(cv_error), mean(auc_score)))
}
```

Here, we do 3-Fold CV to tune hyper paramaters:

```
set.seed(07122021)
K_fold = 3
berr_list = rep(0, nrow(hyper_grid))
berr_min_list = rep(0, nrow(hyper_grid))
berr_max_list = rep(0, nrow(hyper_grid))
auc_list = rep(0, nrow(hyper_grid))
for(row_i in 1:nrow(hyper_grid)){
  result = Kfold_CV_adaboost(K = K_fold, param_grid = hyper_grid, param_combo = row_i, train = val.trai
  berr_list[row_i] = result[1]
  berr_min_list[row_i] = result[2]
  berr_max_list[row_i] = result[3]
  auc_list[row_i] = result[4]
}
```

```
hyper_grid$berr <- berr_list
hyper_grid$berr_min <-berr_min_list
hyper_grid$berr_max <-berr_max_list
hyper_grid$auc <- auc_list
hyper_grid[order(hyper_grid$berr), ]
```

```
##    cutoff n.trees shrinkage interaction.depth      berr  berr_min  berr_max
## 18   0.70    1000      0.01                 4 0.3005548 0.2958542 0.3044565
## 30   0.70    1000      0.01                 5 0.3008687 0.2987302 0.3035746
## 6    0.70    1000      0.01                 3 0.3013312 0.2979190 0.3039694
## 27   0.70     500      0.01                 5 0.3019036 0.2998508 0.3052793
## 21   0.70     500      0.05                 4 0.3061442 0.2982566 0.3133302
## 15   0.70     500      0.01                 4 0.3062246 0.3010762 0.3088948
## 29   0.65    1000      0.01                 5 0.3064155 0.3002378 0.3134567
## 26   0.65     500      0.01                 5 0.3064228 0.3050532 0.3091185
## 3    0.70     500      0.01                 3 0.3081782 0.3045738 0.3106875
## 14   0.65     500      0.01                 4 0.3088620 0.3057458 0.3117202
## 20   0.65     500      0.05                 4 0.3091400 0.3068083 0.3108835
## 17   0.65    1000      0.01                 4 0.3096382 0.3057469 0.3117219
## 5    0.65    1000      0.01                 3 0.3100143 0.3066754 0.3132326
## 2    0.65     500      0.01                 3 0.3109751 0.3084001 0.3154343
## 33   0.70     500      0.05                 5 0.3111919 0.3077047 0.3159764
## 9    0.70     500      0.05                 3 0.3113292 0.3064920 0.3143207
## 16   0.60    1000      0.01                 4 0.3120107 0.3070824 0.3151891
## 32   0.65     500      0.05                 5 0.3135907 0.3065491 0.3209417
## 4    0.60    1000      0.01                 3 0.3140302 0.3119970 0.3157597
## 25   0.60     500      0.01                 5 0.3170032 0.3125538 0.3223064
## 8    0.65     500      0.05                 3 0.3173147 0.3072311 0.3272868
## 24   0.70    1000      0.05                 4 0.3178184 0.3046988 0.3296700
## 28   0.60    1000      0.01                 5 0.3180581 0.3107081 0.3271613
## 1    0.60     500      0.01                 3 0.3187432 0.3099542 0.3270438
```

```
## 13   0.60     500      0.01                4 0.3189351 0.3152330 0.3218694
## 11   0.65    1000      0.05                3 0.3195752 0.3170371 0.3241915
## 7    0.60     500      0.05                3 0.3214429 0.3130167 0.3292041
## 12   0.70    1000      0.05                3 0.3216016 0.3100767 0.3298872
## 31   0.60     500      0.05                5 0.3224151 0.3143799 0.3265708
## 10   0.60    1000      0.05                3 0.3229275 0.3153080 0.3333510
## 36   0.70    1000      0.05                5 0.3237503 0.3150437 0.3315514
## 23   0.65    1000      0.05                4 0.3239293 0.3133296 0.3350230
## 19   0.60     500      0.05                4 0.3253242 0.3147664 0.3331652
## 35   0.65    1000      0.05                5 0.3257915 0.3236208 0.3293900
## 34   0.60    1000      0.05                5 0.3298131 0.3192294 0.3434399
## 22   0.60    1000      0.05                4 0.3309044 0.3220569 0.3407477
##          auc
## 18 0.7662324
## 30 0.7643645
## 6  0.7645486
## 27 0.7632030
## 21 0.7607336
## 15 0.7628255
## 29 0.7666584
## 26 0.7636850
## 3  0.7611773
## 14 0.7619114
## 20 0.7591751
## 17 0.7635774
## 5  0.7618913
## 2  0.7602581
## 33 0.7597333
## 9  0.7581840
## 16 0.7645602
## 32 0.7588707
## 4  0.7637315
## 25 0.7630872
## 8  0.7574515
## 24 0.7560035
## 28 0.7652581
## 1  0.7592888
## 13 0.7615369
## 11 0.7558004
## 7  0.7609547
## 12 0.7521820
## 31 0.7564141
## 10 0.7537629
## 36 0.7511981
## 23 0.7526621
## 19 0.7605255
## 35 0.7541139
## 34 0.7549065
## 22 0.7517562
```

The Best Model

```
# Model with Lowest BER
hyper_grid[which.min(hyper_grid$berr), ]
```

```
##    cutoff n.trees shrinkage interaction.depth      berr  berr_min  berr_max
## 18    0.7    1000      0.01                 4 0.3005548 0.2958542 0.3044565
##          auc
## 18 0.7662324
```

```
#Model with Highest AUC
hyper_grid[which.max(hyper_grid$auc), ]
```

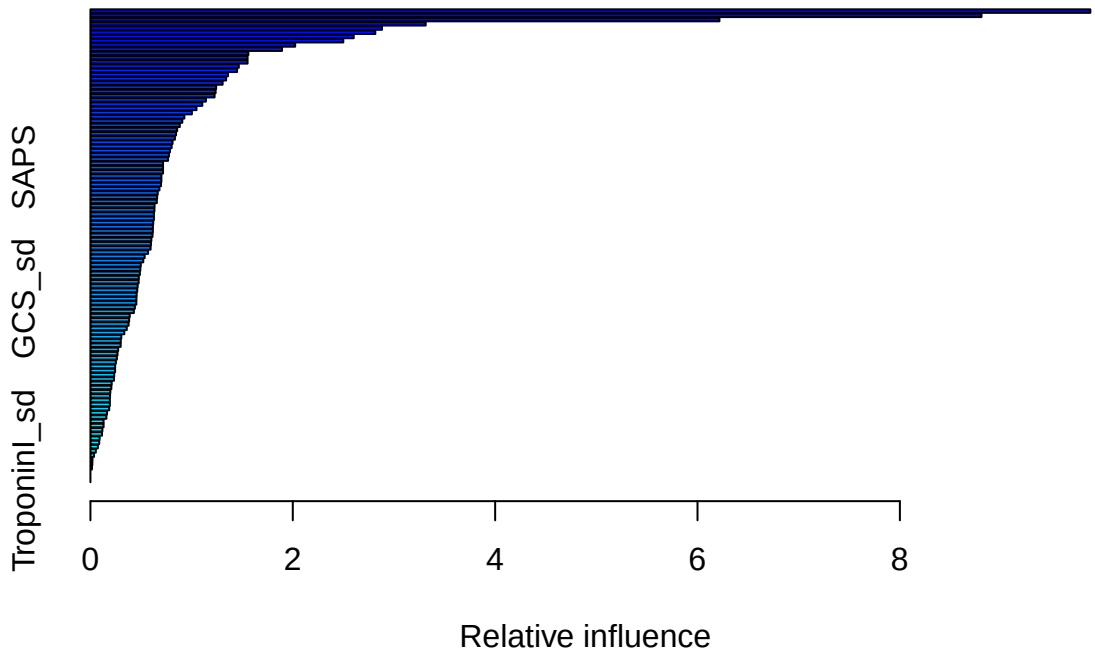
```
##    cutoff n.trees shrinkage interaction.depth      berr  berr_min  berr_max
## 29   0.65    1000      0.01                 5 0.3064155 0.3002378 0.3134567
##          auc
## 29 0.7666584
```

Validating "best" model using validation test set

```
set.seed(08012021)

# Get best values (based on lowest BERR score)
best_cutoff = hyper_grid[which.min(hyper_grid$berr), ]$cutoff
best_n.tress = hyper_grid[which.min(hyper_grid$berr), ]$n.trees
best_interaction.depth = hyper_grid[which.min(hyper_grid$berr), ]$interaction.depth
best_shrinkage = hyper_grid[which.min(hyper_grid$berr), ]$shrinkage

# Fit model on entire training set
ada_best = gbm(outcome ~., data = val.train[, -c(1, 5)], distribution = "adaboost", n.trees = best_n.tr
summary(ada_best)
```




```
##                                    var    rel.inf
## PaO2_reg_coeff             PaO2_reg_coeff 9.88271779
## GCS_median                     GCS_median 8.80668701
## GCS_reg_coeff               GCS_reg_coeff 6.21649544
## Urine_median                 Urine_median 3.31340175
## HR_median                       HR_median 2.88185895
## AdmissionType               AdmissionType 2.81554062
## BUN_median                     BUN_median 2.60209850
## HCT_median                     HCT_median 2.49896303
## Creatinine_median       Creatinine_median 2.02358958
## Urine_sd                         Urine_sd 1.89317844
## Albumin_median             Albumin_median 1.56087666
## NIMAP_sd                         NIMAP_sd 1.55300245
## Na_median                       Na_median 1.55296691
## SysABP_reg_coeff         SysABP_reg_coeff 1.46639924
## WBC_reg_coeff               WBC_reg_coeff 1.44919046
## BUN_reg_coeff               BUN_reg_coeff 1.35954391
## RespRate_median           RespRate_median 1.34095152
## Age                                 Age 1.30651611
## DiasABP_median             DiasABP_median 1.24281633
## Platelets_median         Platelets_median 1.23708422
## FiO2_reg_coeff               FiO2_reg_coeff 1.22684155
## DiasABP_reg_coeff       DiasABP_reg_coeff 1.14114463
## SysABP_sd                       SysABP_sd 1.10567798
## Cholesterol_median     Cholesterol_median 1.04812901
## RespRate_reg_coeff     RespRate_reg_coeff 1.00260236
```

```
## SaO2_median                    SaO2_median 0.92818372
## PaO2_median                    PaO2_median 0.90793859
## PaO2_sd                            PaO2_sd 0.88349200
## MechVent_reg_coeff       MechVent_reg_coeff 0.85747248
## NIDiasABP_sd                  NIDiasABP_sd 0.84573676
## RespRate_sd                    RespRate_sd 0.83499875
## HCT_sd                              HCT_sd 0.81166682
## NIMAP_median                  NIMAP_median 0.80258247
## HCO3_median                    HCO3_median 0.78580966
## NIDiasABP_reg_coeff     NIDiasABP_reg_coeff 0.77589862
## Glucose_median               Glucose_median 0.76751679
## HR_sd                                HR_sd 0.71785751
## SAPS                                SAPS 0.71768494
## pH_reg_coeff                  pH_reg_coeff 0.71756038
## Temp_sd                            Temp_sd 0.70156040
## WBC_median                      WBC_median 0.70130314
## Creatinine_reg_coeff   Creatinine_reg_coeff 0.69752969
## Platelets_sd                  Platelets_sd 0.68201137
## MAP_reg_coeff                MAP_reg_coeff 0.66520963
## Glucose_reg_coeff        Glucose_reg_coeff 0.65941802
## HCO3_sd                            HCO3_sd 0.65629534
## K_reg_coeff                    K_reg_coeff 0.63355187
## Mg_median                        Mg_median 0.63269067
## Urine_reg_coeff            Urine_reg_coeff 0.62766090
## Platelets_reg_coeff     Platelets_reg_coeff 0.62759724
## FiO2_sd                            FiO2_sd 0.62083478
## Lactate_reg_coeff        Lactate_reg_coeff 0.61747022
## NIDiasABP_median          NIDiasABP_median 0.61711855
## pH_sd                                pH_sd 0.61364205
## K_sd                                  K_sd 0.60231691
## SaO2_sd                            SaO2_sd 0.59831885
## Mg_sd                                Mg_sd 0.59457536
## MechVent_sd                    MechVent_sd 0.56956200
## Na_reg_coeff                  Na_reg_coeff 0.53608675
## NISysABP_sd                    NISysABP_sd 0.52211543
## FiO2_median                    FiO2_median 0.49770872
## HCT_reg_coeff                HCT_reg_coeff 0.49330824
## Creatinine_sd                Creatinine_sd 0.48876841
## NISysABP_median            NISysABP_median 0.47836192
## MAP_sd                              MAP_sd 0.47658974
## HCO3_reg_coeff              HCO3_reg_coeff 0.46673124
## MechVent_median            MechVent_median 0.46250324
## Na_sd                                Na_sd 0.45709215
## GCS_sd                              GCS_sd 0.45451661
## NISysABP_reg_coeff       NISysABP_reg_coeff 0.45375307
## TroponinT_reg_coeff     TroponinT_reg_coeff 0.43942951
## BUN_sd                              BUN_sd 0.42912531
## PaCO2_reg_coeff            PaCO2_reg_coeff 0.38858729
## PaCO2_sd                          PaCO2_sd 0.38279404
## Mg_reg_coeff                  Mg_reg_coeff 0.37681230
## Lactate_median              Lactate_median 0.35930997
## DiasABP_sd                      DiasABP_sd 0.33506572
## PaCO2_median                  PaCO2_median 0.30521583
## Temp_median                    Temp_median 0.29987426
```

```
## Glucose_sd                       Glucose_sd 0.29890375
## K_median                           K_median 0.27400826
## WBC_sd                               WBC_sd 0.26772654
## Bilirubin_median           Bilirubin_median 0.26138053
## HR_reg_coeff                   HR_reg_coeff 0.25135942
## TroponinT_sd                   TroponinT_sd 0.24442187
## MAP_median                       MAP_median 0.24415501
## SaO2_reg_coeff               SaO2_reg_coeff 0.23520027
## Bilirubin_sd                   Bilirubin_sd 0.23234171
## Temp_reg_coeff               Temp_reg_coeff 0.21008256
## SysABP_median                 SysABP_median 0.20670668
## ALT_median                       ALT_median 0.19666430
## pH_median                         pH_median 0.19390086
## Albumin_sd                       Albumin_sd 0.19220033
## NIMAP_reg_coeff             NIMAP_reg_coeff 0.19203767
## TroponinT_median           TroponinT_median 0.18562716
## AST_sd                               AST_sd 0.16370900
## ALP_sd                               ALP_sd 0.15787695
## ALT_reg_coeff                 ALT_reg_coeff 0.13023884
## Bilirubin_reg_coeff     Bilirubin_reg_coeff 0.13015379
## ALP_median                       ALP_median 0.11800270
## Lactate_sd                       Lactate_sd 0.11439504
## ALP_reg_coeff                 ALP_reg_coeff 0.09088282
## Albumin_reg_coeff         Albumin_reg_coeff 0.08664103
## AST_median                       AST_median 0.07341573
## TroponinI_median           TroponinI_median 0.05323147
## AST_reg_coeff                 AST_reg_coeff 0.03565738
## ALT_sd                               ALT_sd 0.02040531
## Gender                               Gender 0.01882626
## TroponinI_reg_coeff     TroponinI_reg_coeff 0.01475610
## Cholesterol_reg_coeff Cholesterol_reg_coeff 0.00000000
## Cholesterol_sd               Cholesterol_sd 0.00000000
## TroponinI_sd                   TroponinI_sd 0.00000000
```

```
#Make predictions using validation test set
ada_pred_response = predict(ada_best, newdata = val.test[, -c(1, 5)], n.trees = best_n.tress, type = "r
ada_pred = ifelse(ada_pred_response > best_cutoff, 1, 0)
pos_error <- mean(ada_pred[which(val.test$outcome == 1)] != val.test[val.test$outcome == 1,]$outcome)
neg_error <- mean(ada_pred[which(val.test$outcome == 0)] != val.test[val.test$outcome == 0, ]$outcome)

# Calculate balanced error rate
berr_error = (pos_error + neg_error) / 2
sprintf("Balanced Error Rate based on Validation Test Set: %f", berr_error)
```

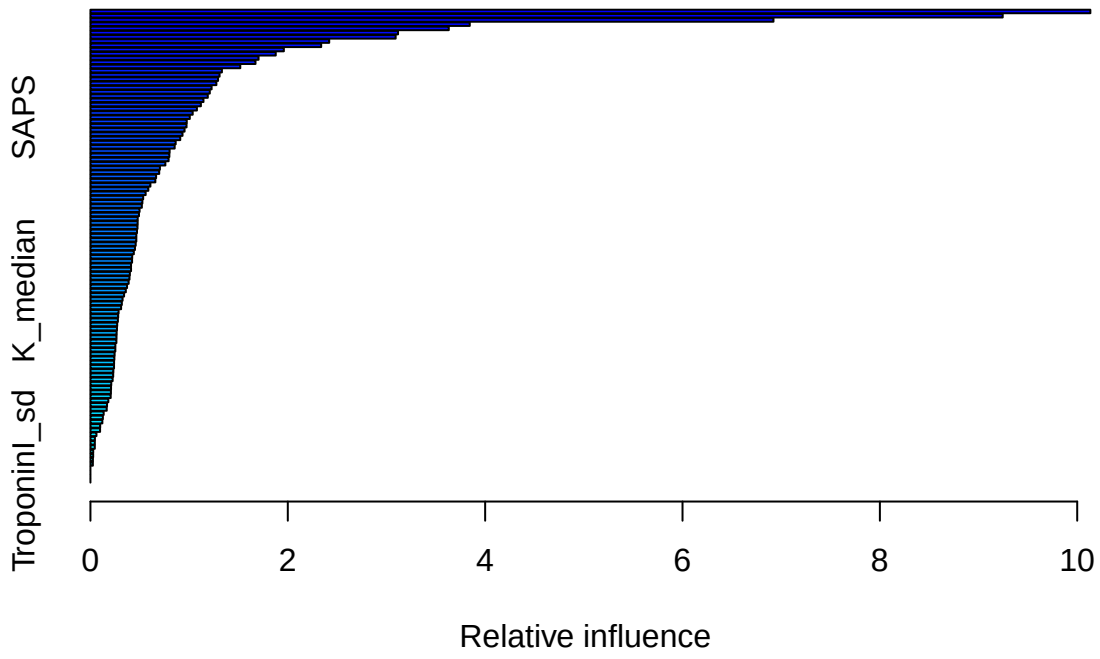
```
## [1] "Balanced Error Rate based on Validation Test Set: 0.299093"
```

```
# Calculate AUC score
pr <- prediction(ada_pred_response, val.test$outcome)
auc = performance(pr, "auc")
sprintf("AUC score based on Validation Test Set: %f", as.numeric(auc@y.values))
```

```
## [1] "AUC score based on Validation Test Set: 0.772792"
```

Final Predictions

```
# Re-Fit model on entire training set
final_ada_model = gbm(outcome ~., data = train_data.sumvars[, -c(1, 5)], distribution = "adaboost", n.t
summary(final_ada_model)
```




```
##                                    var     rel.inf
## GCS_median                   GCS_median 10.13310144
## PaO2_reg_coeff         PaO2_reg_coeff  9.24473985
## GCS_reg_coeff           GCS_reg_coeff  6.92176447
## AdmissionType           AdmissionType  3.84418468
## Urine_median             Urine_median  3.63157778
## BUN_median                 BUN_median  3.11721479
## HR_median                   HR_median  3.09319718
## Urine_sd                     Urine_sd  2.41967457
## HCT_median                 HCT_median  2.33825351
## PaO2_sd                       PaO2_sd  1.96011067
## NIMAP_sd                     NIMAP_sd  1.87780238
## WBC_reg_coeff           WBC_reg_coeff  1.70381525
## SysABP_reg_coeff     SysABP_reg_coeff  1.67277515
## RespRate_median       RespRate_median  1.51848039
## Age                               Age  1.33239917
## BUN_reg_coeff           BUN_reg_coeff  1.30871345
## Creatinine_median   Creatinine_median  1.29336715
```

```
## Na_median                       Na_median  1.27640575
## DiasABP_reg_coeff         DiasABP_reg_coeff  1.22803666
## Platelets_median           Platelets_median  1.20969114
## Albumin_median               Albumin_median  1.18920446
## RespRate_reg_coeff       RespRate_reg_coeff  1.14547560
## MechVent_reg_coeff       MechVent_reg_coeff  1.11942545
## DiasABP_median               DiasABP_median  1.07913194
## Glucose_median               Glucose_median  1.03639597
## SAPS                                   SAPS  1.00681779
## NIDiasABP_sd                 NIDiasABP_sd  0.97603750
## HCO3_median                   HCO3_median  0.97386196
## FiO2_reg_coeff               FiO2_reg_coeff  0.95383005
## Platelets_sd                   Platelets_sd  0.93321838
## Bilirubin_median           Bilirubin_median  0.91011306
## MechVent_sd                     MechVent_sd  0.86390148
## Glucose_reg_coeff         Glucose_reg_coeff  0.85370311
## NIMAP_median                   NIMAP_median  0.80252198
## Cholesterol_median       Cholesterol_median  0.79993270
## FiO2_sd                             FiO2_sd  0.79226148
## RespRate_sd                     RespRate_sd  0.75946677
## NIDiasABP_median           NIDiasABP_median  0.70418365
## PaO2_median                     PaO2_median  0.69682218
## Lactate_reg_coeff         Lactate_reg_coeff  0.66569235
## WBC_median                       WBC_median  0.65775599
## SysABP_sd                         SysABP_sd  0.60783769
## Mg_reg_coeff                   Mg_reg_coeff  0.58687218
## Urine_reg_coeff             Urine_reg_coeff  0.55955114
## pH_median                         pH_median  0.53444796
## Na_sd                                 Na_sd  0.52640333
## Creatinine_reg_coeff   Creatinine_reg_coeff  0.52102074
## WBC_sd                               WBC_sd  0.49805014
## Platelets_reg_coeff     Platelets_reg_coeff  0.49189771
## HCT_sd                               HCT_sd  0.47914670
## K_sd                                   K_sd  0.47849427
## MAP_reg_coeff                 MAP_reg_coeff  0.47701943
## TroponinT_sd                   TroponinT_sd  0.47178617
## Creatinine_sd                 Creatinine_sd  0.46543071
## Glucose_sd                       Glucose_sd  0.46512464
## HR_sd                                 HR_sd  0.45883646
## Mg_median                         Mg_median  0.45129636
## MechVent_median             MechVent_median  0.43961081
## pH_reg_coeff                   pH_reg_coeff  0.42308673
## NISysABP_sd                     NISysABP_sd  0.42223645
## DiasABP_sd                       DiasABP_sd  0.41153029
## HCO3_reg_coeff               HCO3_reg_coeff  0.41146870
## NIDiasABP_reg_coeff     NIDiasABP_reg_coeff  0.39967349
## Temp_reg_coeff               Temp_reg_coeff  0.39591057
## HCT_reg_coeff                 HCT_reg_coeff  0.38734090
## HR_reg_coeff                   HR_reg_coeff  0.37107660
## K_median                           K_median  0.35853031
## HCO3_sd                             HCO3_sd  0.34111424
## SaO2_sd                             SaO2_sd  0.32603575
## NISysABP_median             NISysABP_median  0.31824782
## Temp_sd                             Temp_sd  0.31063921
```

```
## BUN_sd                             BUN_sd  0.28545336
## PaCO2_reg_coeff           PaCO2_reg_coeff  0.28136008
## Mg_sd                               Mg_sd  0.27830739
## SaO2_median                   SaO2_median  0.27122557
## pH_sd                               pH_sd  0.26890627
## Na_reg_coeff                 Na_reg_coeff  0.26564898
## SysABP_median               SysABP_median  0.26432150
## Lactate_median             Lactate_median  0.26322259
## Bilirubin_sd                 Bilirubin_sd  0.25268798
## GCS_sd                             GCS_sd  0.25251137
## MAP_median                   MAP_median  0.24512307
## ALP_median                   ALP_median  0.24272955
## Albumin_sd                   Albumin_sd  0.23968756
## NISysABP_reg_coeff     NISysABP_reg_coeff  0.23913149
## PaCO2_median                 PaCO2_median  0.23250949
## MAP_sd                             MAP_sd  0.23084926
## NIMAP_reg_coeff           NIMAP_reg_coeff  0.22501892
## AST_median                   AST_median  0.21114404
## FiO2_median                 FiO2_median  0.20884063
## ALT_median                   ALT_median  0.20756580
## PaCO2_sd                       PaCO2_sd  0.20528254
## TroponinT_reg_coeff   TroponinT_reg_coeff  0.18148212
## K_reg_coeff                 K_reg_coeff  0.16834602
## SaO2_reg_coeff           SaO2_reg_coeff  0.16498573
## Lactate_sd                   Lactate_sd  0.13438364
## Temp_median                 Temp_median  0.12625963
## AST_sd                             AST_sd  0.12077220
## Bilirubin_reg_coeff   Bilirubin_reg_coeff  0.09884285
## ALT_sd                             ALT_sd  0.09776181
## ALP_sd                             ALP_sd  0.06114144
## TroponinT_median         TroponinT_median  0.04529587
## ALT_reg_coeff               ALT_reg_coeff  0.04469620
## Albumin_reg_coeff       Albumin_reg_coeff  0.04436606
## ALP_reg_coeff               ALP_reg_coeff  0.02810012
## AST_reg_coeff               AST_reg_coeff  0.02774142
## TroponinI_median         TroponinI_median  0.02504636
## TroponinI_reg_coeff   TroponinI_reg_coeff  0.02447431
## Gender                           Gender  0.00000000
## Cholesterol_reg_coeff Cholesterol_reg_coeff  0.00000000
## Cholesterol_sd             Cholesterol_sd  0.00000000
## TroponinI_sd                 TroponinI_sd  0.00000000
```

```
#Make predictions using Test Set
final_ada_prob = predict(final_ada_model, newdata = test_data.sumvars[, -c(1, 5)], n.trees = best_n.tres
final_ada_pred = ifelse(final_ada_prob > best_cutoff, 1, 0)

# Store Results
test_outcomes$score <- final_ada_prob
test_outcomes$outcome <- final_ada_pred
head(test_outcomes)
```

```
##     id outcome     score
## 1 7090       0 0.5769003
## 2 7091       1 0.8187937
```

```
## 3 7092       1 0.8731926
## 4 7093       1 0.9046622
## 5 7094       0 0.6353587
## 6 7095       1 0.9271459
```

Save Results

```
write.csv(test_outcomes,"../Final Model and Predictions/test_nolabel.csv", row.names = FALSE)
```