Titanic Analysis

Jerry Kiely

14 February 2015

The Introduction

Load the cleaned data:

```
train <- load_training_data();
test  <- load_testing_data();

str(train);
```

```
## 'data.frame':    891 obs. of  7 variables:
##  $ Survived  : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass    : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex       : chr  "male" "female" "female" "female" ...
##  $ Age       : num  22 38 26 35 35 28 54 2 27 14 ...
##  $ Embarked  : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  $ Title     : Factor w/ 6 levels "Lady","Master",..: 4 5 3 5 4 4 4 2 5 5 ...
##  $ FamilySize: num  2 2 1 2 1 1 1 5 3 2 ...
```

The Analysis

Partition the data into training and cross validation:

```
partition <- createDataPartition(y = train$Survived, p = 0.75, list = FALSE);
part_tr   <- train[ partition, ];
part_cv   <- train[-partition, ];
```

Train the Random Forest model:

```
model1 <- train(
    Survived ~ .,
    method     = 'rf',
    data       = part_tr,
    importance = TRUE,
    trControl  = trainControl(method = 'oob', number = 4)
);
```

Now train the glm model:

```
model2 <- train(
    Survived ~ .,
    method     = 'glm',
    data       = part_tr,
    family     = binomial
);
```

Now train the gbm model:

```
    model3 <- train(
        Survived ~ .,
        method      = 'gbm',
        data        = part_tr,
        verbose     = FALSE
    );
```

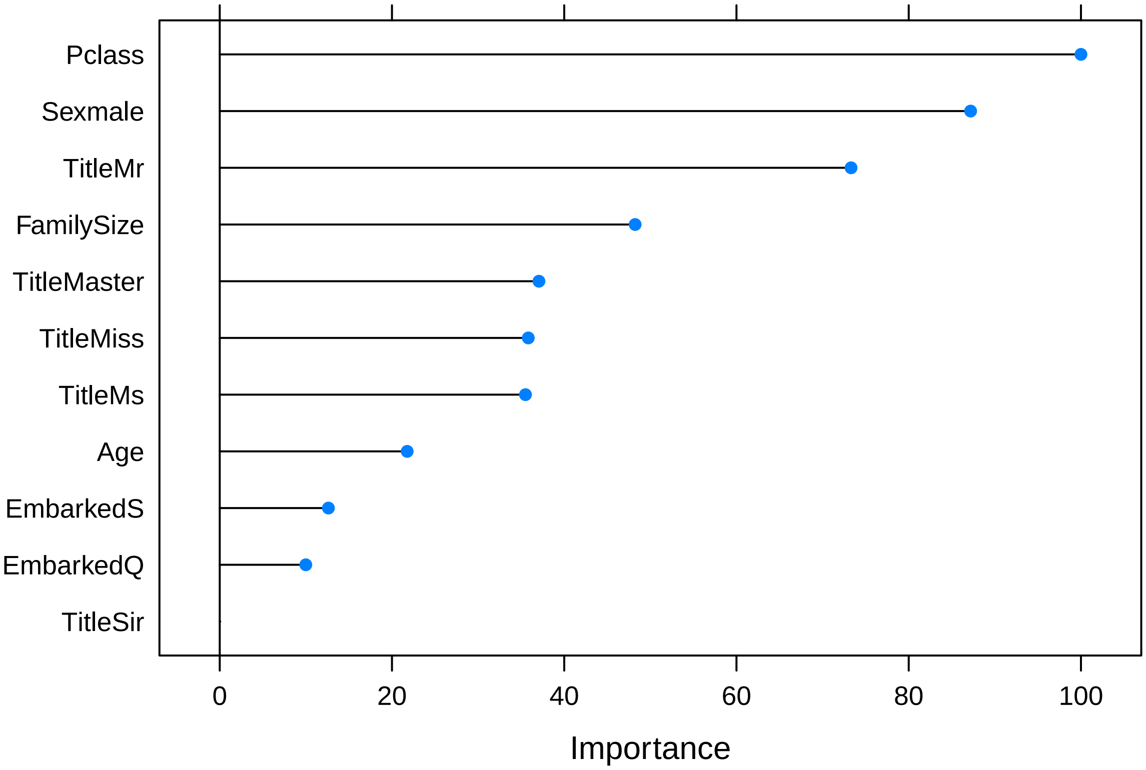
Look at the importance of the various features / variables of the Random Forest model:

```
     importance1 <- varImp(model1);

     importance1;
```

```
## rf variable importance
##
##               Importance
## Pclass           100.000
## Sexmale           87.186
## TitleMr           73.306
## FamilySize        48.235
## TitleMaster       37.065
## TitleMiss         35.829
## TitleMs           35.501
## Age               21.766
## EmbarkedS         12.622
## EmbarkedQ          9.994
## TitleSir           0.000
```

```
     plot(importance1);
```



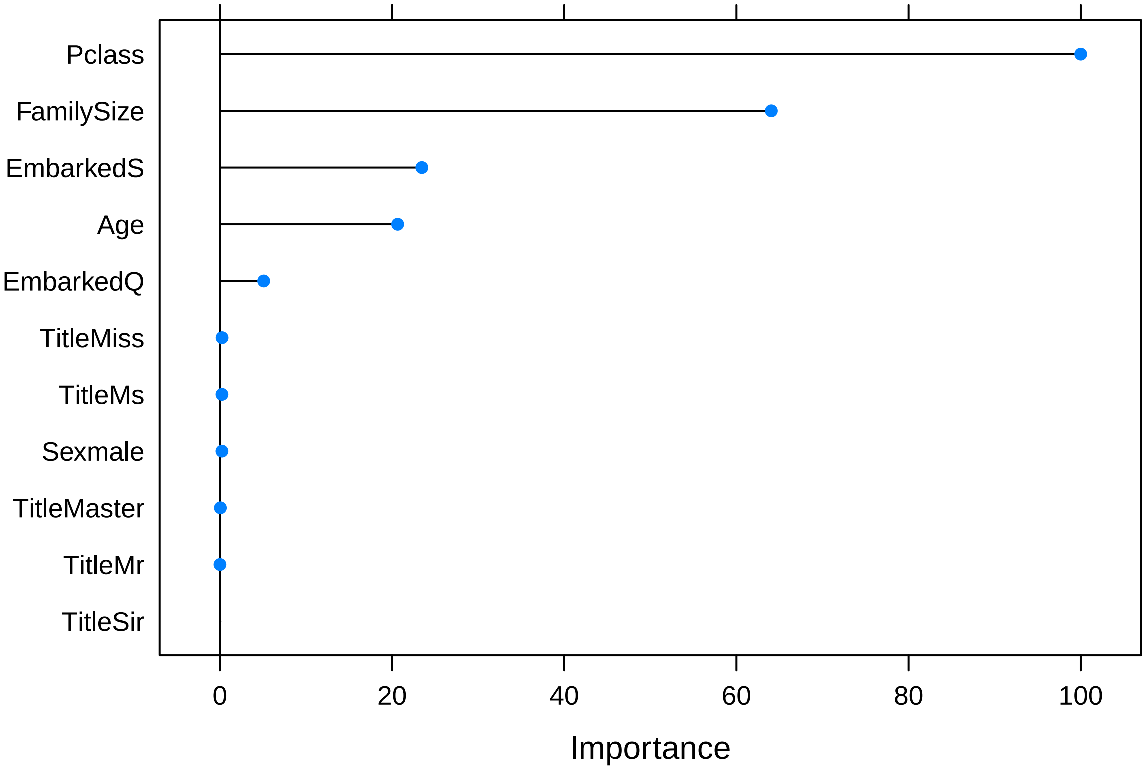
Look at the importance of the various features / variables of the glm model:

```
importance2 <- varImp(model2);

importance2;
```

```
## glm variable importance
##
##               Overall
## Pclass      1.000e+02
## FamilySize  6.405e+01
## EmbarkedS   2.346e+01
## Age         2.065e+01
## EmbarkedQ   5.091e+00
## TitleMiss   2.527e-01
## TitleMs     2.386e-01
## Sexmale     2.350e-01
## TitleMaster 5.556e-02
## TitleMr     7.092e-03
## TitleSir    0.000e+00
```

```
plot(importance2);
```



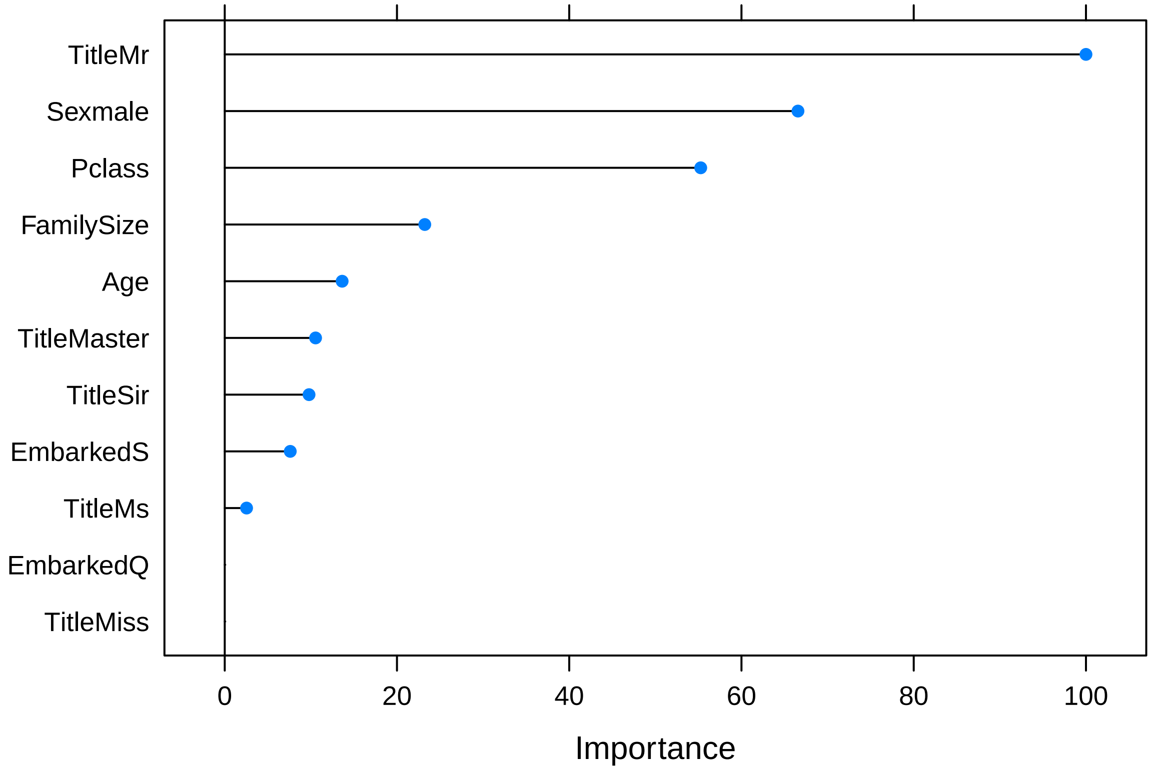
Look at the importance of the various features / variables of the gbm model:

```
importance3 <- varImp(model3);

importance3;
```

```
## gbm variable importance
##
##             Overall
## TitleMr     100.000
## Sexmale      66.561
## Pclass       55.270
## FamilySize   23.231
## Age          13.634
## TitleMaster  10.549
## TitleSir      9.789
## EmbarkedS     7.608
## TitleMs       2.537
## TitleMiss     0.000
## EmbarkedQ     0.000
```

```
plot(importance3);
```



Look at the final model of the Random Forest:

```
model1;
```

```
## Random Forest
##
## 669 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8340807  0.6443926
##    6    0.8086697  0.5828080
##   11    0.7922272  0.5560081
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

```
model1$finalModel
```

```
##
## Call:
```

```
##  randomForest(x = x, y = y, mtry = param$mtry, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.59%
## Confusion matrix:
##     0   1 class.error
## 0 366  46   0.1116505
## 1  65 192   0.2529183
```

Look at the final model of the glm:

```
model2;
```

```
## Generalized Linear Model
##
## 669 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 669, 669, 669, 669, 669, 669, ...
##
## Resampling results
##
##   Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.8225105  0.6177639  0.02119525   0.04387405
##
##
```

```
model2$finalModel
```

```
##
## Call:  NULL
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age    EmbarkedQ
##    17.26807     -1.16180    -15.73825     -0.01576     -0.17325
##   EmbarkedS  TitleMaster    TitleMiss      TitleMr      TitleMs
##    -0.48115      4.67112    -11.95028      0.72887    -11.28558
##    TitleSir   FamilySize
##     0.15202     -0.43869
##
## Degrees of Freedom: 668 Total (i.e. Null);  657 Residual
## Null Deviance:       891.2
## Residual Deviance: 538.8     AIC: 562.8
```

Look at the final model of the gbm:

```
    model3;
```

```
## Stochastic Gradient Boosting
##
## 669 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 669, 669, 669, 669, 669, 669, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.8184601  0.6131508  0.01933080
##   1                  100      0.8233412  0.6231080  0.01581182
##   1                  150      0.8198261  0.6158445  0.01689269
##   2                   50      0.8227302  0.6188773  0.01957513
##   2                  100      0.8210930  0.6164586  0.01830306
##   2                  150      0.8187520  0.6113492  0.02047244
##   3                   50      0.8209473  0.6140970  0.02117424
##   3                  100      0.8165301  0.6071186  0.02098176
##   3                  150      0.8164603  0.6070321  0.02291753
##   Kappa SD
##   0.04297312
##   0.03524818
##   0.03781943
##   0.04364883
##   0.04135438
##   0.04643352
##   0.04522937
##   0.04559926
##   0.04835971
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 100,
##  interaction.depth = 1 and shrinkage = 0.1.
```

```
    model3$finalModel
```

```
## A gradient boosted model with bernoulli loss function.
## 100 iterations were performed.
## There were 11 predictors of which 9 had non-zero influence.
```

Predict with the training set using the Random Forest model:

```
    predict1_tr <- predict(model1, part_tr);
    cm_tr       <- confusionMatrix(predict1_tr, part_tr$Survived);
    cm_tr;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 370  64
##          1  42 193
##
##                Accuracy : 0.8416
##                  95% CI : (0.8116, 0.8684)
##     No Information Rate : 0.6158
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6597
##  Mcnemar's Test P-Value : 0.04138
##
##             Sensitivity : 0.8981
##             Specificity : 0.7510
##          Pos Pred Value : 0.8525
##          Neg Pred Value : 0.8213
##              Prevalence : 0.6158
##          Detection Rate : 0.5531
##    Detection Prevalence : 0.6487
##       Balanced Accuracy : 0.8245
##
##        'Positive' Class : 0
##
```

Predict with the cross validation set using the Random Forest model:

```
    predict1_cv <- predict(model1, part_cv);
    cm_cv       <- confusionMatrix(predict1_cv, part_cv$Survived);
    cm_cv;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 125  26
##          1  12  59
##
##                Accuracy : 0.8288
##                  95% CI : (0.7727, 0.8759)
##     No Information Rate : 0.6171
##     P-Value [Acc > NIR] : 5.476e-12
##
##                   Kappa : 0.6261
##  Mcnemar's Test P-Value : 0.03496
##
##             Sensitivity : 0.9124
##             Specificity : 0.6941
##          Pos Pred Value : 0.8278
##          Neg Pred Value : 0.8310
##              Prevalence : 0.6171
##          Detection Rate : 0.5631
##    Detection Prevalence : 0.6802
##       Balanced Accuracy : 0.8033
##
##        'Positive' Class : 0
##
```

Predict with the training set using the glm model:

```
    predict2_tr <- predict(model2, part_tr);
    cm_tr       <- confusionMatrix(predict2_tr, part_tr$Survived);
    cm_tr;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 362  64
##          1  50 193
##
##                Accuracy : 0.8296
##                  95% CI : (0.7989, 0.8573)
##     No Information Rate : 0.6158
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6361
##  Mcnemar's Test P-Value : 0.2234
##
##             Sensitivity : 0.8786
##             Specificity : 0.7510
##          Pos Pred Value : 0.8498
##          Neg Pred Value : 0.7942
##              Prevalence : 0.6158
##          Detection Rate : 0.5411
##    Detection Prevalence : 0.6368
##       Balanced Accuracy : 0.8148
##
##        'Positive' Class : 0
##
```

Predict with the cross validation set using the glm model:

```
    predict2_cv <- predict(model2, part_cv);
    cm_cv       <- confusionMatrix(predict2_cv, part_cv$Survived);
    cm_cv;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 125  25
##          1  12  60
```

```
##
##                Accuracy : 0.8333
##                  95% CI : (0.7777, 0.8799)
##     No Information Rate : 0.6171
##     P-Value [Acc > NIR] : 1.786e-12
##
##                   Kappa : 0.6368
##  Mcnemar's Test P-Value : 0.04852
##
##             Sensitivity : 0.9124
##             Specificity : 0.7059
##          Pos Pred Value : 0.8333
##          Neg Pred Value : 0.8333
##              Prevalence : 0.6171
##          Detection Rate : 0.5631
##    Detection Prevalence : 0.6757
##       Balanced Accuracy : 0.8091
##
##        'Positive' Class : 0
##
```

Predict with the training set using the gbm model:

```
    predict3_tr <- predict(model3, part_tr);
    cm_tr       <- confusionMatrix(predict3_tr, part_tr$Survived);
    cm_tr;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 366  64
##          1  46 193
##
##                Accuracy : 0.8356
##                  95% CI : (0.8053, 0.8629)
##     No Information Rate : 0.6158
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6479
##  Mcnemar's Test P-Value : 0.105
##
##             Sensitivity : 0.8883
##             Specificity : 0.7510
##          Pos Pred Value : 0.8512
##          Neg Pred Value : 0.8075
##              Prevalence : 0.6158
##          Detection Rate : 0.5471
##    Detection Prevalence : 0.6428
##       Balanced Accuracy : 0.8197
##
##        'Positive' Class : 0
##
```

Predict with the cross validation set using the gbm model:

```
    predict3_cv <- predict(model3, part_cv);
    cm_cv       <- confusionMatrix(predict3_cv, part_cv$Survived);
    cm_cv;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 125  27
##          1  12  58
##
##                Accuracy : 0.8243
##                  95% CI : (0.7678, 0.872)
##     No Information Rate : 0.6171
##     P-Value [Acc > NIR] : 1.628e-11
##
##                   Kappa : 0.6154
##  Mcnemar's Test P-Value : 0.02497
##
##             Sensitivity : 0.9124
##             Specificity : 0.6824
##          Pos Pred Value : 0.8224
##          Neg Pred Value : 0.8286
##              Prevalence : 0.6171
##          Detection Rate : 0.5631
##    Detection Prevalence : 0.6847
##       Balanced Accuracy : 0.7974
##
##        'Positive' Class : 0
##
```

Fit a model that includes all predictors:

```
    pred_tr <- data.frame(
        prediction1 = predict1_tr,
        prediction2 = predict2_tr,
        prediction3 = predict3_tr,
        Survived    = part_tr$Survived
    );

    pred_cv <- data.frame(
        prediction1 = predict1_cv,
        prediction2 = predict2_cv,
        prediction3 = predict3_cv
    );

    comb_model  <- train(
        Survived ~ .,
        method = 'gamboost',
        data   = pred_tr
    );
```

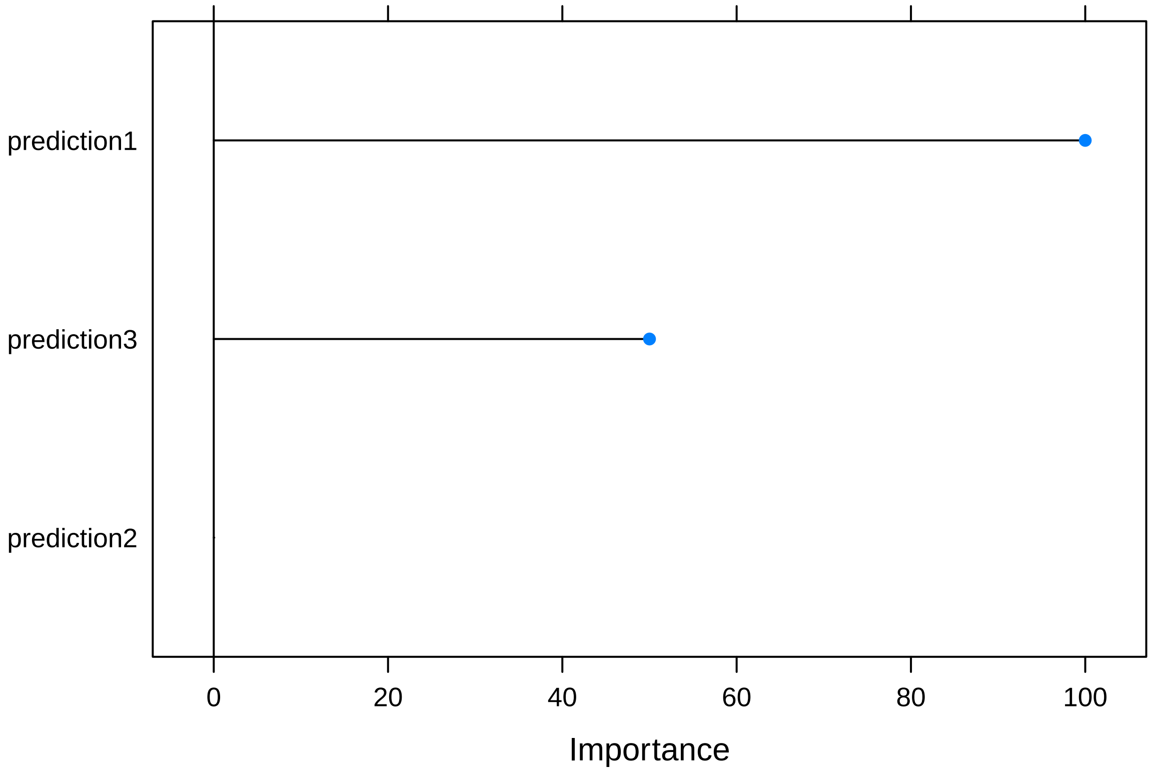
Look at the importance of the various features / variables of the combined model:

```
importance4 <- varImp(comb_model);

importance4;
```

```
## ROC curve variable importance
##
##             Importance
## prediction1        100
## prediction3         50
## prediction2          0
```

```
plot(importance4);
```



Look at the final model of the combined model:

```
comb_model;
```

```
## Boosted Generalized Additive Model
##
## 669 samples
##   3 predictor
##   2 classes: '0', '1'
##
```

```
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 669, 669, 669, 669, 669, 669, ...
##
## Resampling results across tuning parameters:
##
##   mstop  Accuracy   Kappa      Accuracy SD  Kappa SD
##    50    0.8424938  0.6593869  0.02459759   0.05385262
##   100    0.8424938  0.6593869  0.02459759   0.05385262
##   150    0.8424938  0.6593869  0.02459759   0.05385262
##
## Tuning parameter 'prune' was held constant at a value of no
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were mstop = 50 and prune = no.
```

Predict with the training set using the combined model:

```
    predict4_tr <- predict(comb_model, pred_tr);
    cm_tr       <- confusionMatrix(predict4_tr, part_tr$Survived);
    cm_tr;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 370  64
##          1  42 193
##
##                Accuracy : 0.8416
##                  95% CI : (0.8116, 0.8684)
##     No Information Rate : 0.6158
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6597
##  Mcnemar's Test P-Value : 0.04138
##
##             Sensitivity : 0.8981
##             Specificity : 0.7510
##          Pos Pred Value : 0.8525
##          Neg Pred Value : 0.8213
##              Prevalence : 0.6158
##          Detection Rate : 0.5531
##    Detection Prevalence : 0.6487
##       Balanced Accuracy : 0.8245
##
##        'Positive' Class : 0
##
```

Predict with the cross validation set using the combined model:

```r
    predict4_cv <- predict(comb_model, pred_cv);
    cm_cv       <- confusionMatrix(predict4_cv, part_cv$Survived);
    cm_cv;
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 125  26
##          1  12  59
##
##                Accuracy : 0.8288
##                  95% CI : (0.7727, 0.8759)
##     No Information Rate : 0.6171
##     P-Value [Acc > NIR] : 5.476e-12
##
##                   Kappa : 0.6261
##  Mcnemar's Test P-Value : 0.03496
##
##             Sensitivity : 0.9124
##             Specificity : 0.6941
##          Pos Pred Value : 0.8278
##          Neg Pred Value : 0.8310
##              Prevalence : 0.6171
##          Detection Rate : 0.5631
##    Detection Prevalence : 0.6802
##       Balanced Accuracy : 0.8033
##
##        'Positive' Class : 0
##
```