Assignment-4.3—Supervised-Classification-Models.R

arpan

2023-06-07

```r
#Use the attached "titanic.csv" data and do as follows in R Studio with R script:

#1. Read the titanic.csv data with base R function and save it as "data" and
# remove the name column and save again as data
setwd("/Users/arpan/Desktop/MDS/01 MDS I-I/MDS 503 - Statistical Computing with R/Lab/Data")
data <- read.csv("Arpan Sapkota - titanic.csv")
data <- data[, -3]  # Remove the name column
head(data)
```

```
##   Survived Pclass    Sex Age Siblings.Spouses.Aboard Parents.Children.Aboard
## 1        0      3   male  22                       1                       0
## 2        1      1 female  38                       1                       0
## 3        1      3 female  26                       0                       0
## 4        1      1 female  35                       1                       0
## 5        0      3   male  35                       0                       0
## 6        0      3   male  27                       0                       0
##      Fare
## 1  7.2500
## 2 71.2833
## 3  7.9250
## 4 53.1000
## 5  8.0500
## 6  8.4583
```

```r
str(data)
```

```
## 'data.frame':    887 obs. of  7 variables:
##  $ Survived               : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ 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 27 54 2 27 14 ...
##  $ Siblings.Spouses.Aboard: int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parents.Children.Aboard: int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare                   : num  7.25 71.28 7.92 53.1 8.05 ...
```

```r
#Converting factor
table(data$Pclass)
```

```
## 
##   1   2   3 
## 216 184 487
```

```r
data$Pclass<-as.factor(data$Pclass)
str(data$Pclass)
```

```
##  Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
```

```r
data$Sex <- as.factor(data$Sex)
str(data$Sex)
```

```
##  Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
```

```r
data$Survived<-as.factor(data$Survived)
data$Siblings.Spouses.Aboard<-as.factor(data$Siblings.Spouses.Aboard)
data$Parents.Children.Aboard<-as.factor(data$Parents.Children.Aboard)




#2. Fit binary logistic regression model with "Survived" variable as dependent
#variable and rest of variables as independent variables using "data",
# get summary of the model, check VIF and interpret the results carefully
model <- glm(Survived ~ ., data = data, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8978  -0.6112  -0.4012   0.6037   2.4499
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.796e+00  4.841e-01   7.842 4.43e-15 ***
## Pclass2                   -1.184e+00  3.057e-01  -3.874 0.000107 ***
## Pclass3                   -2.211e+00  3.108e-01  -7.114 1.13e-12 ***
## Sexmale                   -2.720e+00  2.023e-01 -13.445  < 2e-16 ***
## Age                       -4.083e-02  8.250e-03  -4.949 7.45e-07 ***
## Siblings.Spouses.Aboard1   2.981e-02  2.254e-01   0.132 0.894785
## Siblings.Spouses.Aboard2  -3.342e-01  5.335e-01  -0.626 0.530997
## Siblings.Spouses.Aboard3  -2.598e+00  7.174e-01  -3.621 0.000293 ***
## Siblings.Spouses.Aboard4  -1.798e+00  7.705e-01  -2.333 0.019633 *
## Siblings.Spouses.Aboard5  -1.612e+01  9.567e+02  -0.017 0.986561
## Siblings.Spouses.Aboard8  -1.666e+01  7.517e+02  -0.022 0.982318
## Parents.Children.Aboard1   3.267e-01  2.897e-01   1.128 0.259432
## Parents.Children.Aboard2   1.667e-02  3.824e-01   0.044 0.965235
## Parents.Children.Aboard3   3.325e-01  1.043e+00   0.319 0.749915
## Parents.Children.Aboard4  -1.601e+01  1.052e+03  -0.015 0.987854
## Parents.Children.Aboard5  -1.217e+00  1.169e+00  -1.041 0.297914
## Parents.Children.Aboard6  -1.656e+01  2.400e+03  -0.007 0.994495
## Fare                       2.787e-03  2.522e-03   1.105 0.269239
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.77  on 886  degrees of freedom
## Residual deviance:  761.48  on 869  degrees of freedom
## AIC: 797.48
##
## Number of Fisher Scoring iterations: 15
```

```
library(car)
```

```
## Loading required package: carData
```

```
vif(model)
```

```
##                             GVIF Df GVIF^(1/(2*Df))
## Pclass                  2.115788  2        1.206058
## Sex                     1.167931  1        1.080709
## Age                     1.582827  1        1.258104
## Siblings.Spouses.Aboard 1.521675  6        1.035603
## Parents.Children.Aboard 1.588688  6        1.039329
## Fare                    1.612524  1        1.269852
```

```
# The VIF values indicate that there is no severe multicollinearity among the
# predictor variables in the logistic regression model. The variables Pclass, Sex,
# Age, Siblings.Spouses.Aboard, Parents.Children.Aboard, and Fare have VIF values
# ranging from 1.167931 to 2.115788 These values suggest that there is little to
# moderate correlation between the predictor variables, indicating that they can
# be included in model without significant issues related to multicollinearity.

#3. Randomly split the data into 70% and 30% with replacement of samples as
#"train" and "test" data
set.seed(07)
ind <- sample(2, nrow(data),replace = T, prob = c(0.7, 0.3))
train <- data[ind==1,]
test <- data[ind==2,]

#4. Fit binary logistic regression classifier, knn classifier, ann classifier,
# naive bayes classifier, svm classifier, decision tree classifier, decision
# tree bagging classifier, random forest classifier, tuned random forest
# classifier and random forest boosting classifier models using the "train" data
library(class)
library(nnet)
library(e1071)
library(rpart)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```r
library(caret)
```

```
## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin

## Loading required package: lattice
```

```r
# Fit binary logistic regression classifier
logit_model <- glm(Survived ~ ., data = train, family = binomial)
summary(logit_model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6101  -0.5823  -0.4181   0.5771   2.3590
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              3.711e+00  5.942e-01   6.246 4.22e-10 ***
## Pclass2                 -1.081e+00  3.808e-01  -2.838 0.004536 **
## Pclass3                 -1.944e+00  3.893e-01  -4.995 5.89e-07 ***
## Sexmale                 -2.869e+00  2.467e-01 -11.630  < 2e-16 ***
## Age                     -3.668e-02  9.901e-03  -3.705 0.000212 ***
## Siblings.Spouses.Aboard1 -3.322e-01  2.866e-01  -1.159 0.246376
## Siblings.Spouses.Aboard2 -1.160e-02  7.025e-01  -0.017 0.986825
## Siblings.Spouses.Aboard3 -2.445e+00  8.612e-01  -2.839 0.004525 **
## Siblings.Spouses.Aboard4 -1.491e+00  8.785e-01  -1.697 0.089665 .
## Siblings.Spouses.Aboard5 -1.660e+01  9.460e+02  -0.018 0.985997
## Siblings.Spouses.Aboard8 -1.720e+01  9.778e+02  -0.018 0.985961
## Parents.Children.Aboard1  2.500e-01  3.562e-01   0.702 0.482843
## Parents.Children.Aboard2  2.528e-01  4.696e-01   0.538 0.590395
## Parents.Children.Aboard3  2.456e-01  1.094e+00   0.225 0.822367
## Parents.Children.Aboard4 -1.615e+01  1.018e+03  -0.016 0.987336
## Parents.Children.Aboard5 -1.428e+00  1.186e+00  -1.204 0.228609
## Parents.Children.Aboard6 -1.663e+01  2.400e+03  -0.007 0.994471
## Fare                     4.343e-03  3.892e-03   1.116 0.264553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 823.16  on 608  degrees of freedom
## Residual deviance: 518.77  on 591  degrees of freedom
```

```
## AIC: 554.77
##
## Number of Fisher Scoring iterations: 15
```

```
library(class)
# Separate predictor variables (train_x) and the class variable (train_y)
train_x <- train[, -1]
train_y <- train$Survived

sum(is.na(train_x))
```

```
## [1] 0
```

```
str(train_x)
```

```
## 'data.frame':    609 obs. of  6 variables:
##  $ Pclass                 : Factor w/ 3 levels "1","2","3": 1 3 1 3 1 3 2 3 1 3 ...
##  $ Sex                    : Factor w/ 2 levels "female","male": 1 1 1 2 2 1 1 1 1 2 ...
##  $ Age                    : num  38 26 35 35 54 27 14 4 58 39 ...
##  $ Siblings.Spouses.Aboard: Factor w/ 7 levels "0","1","2","3",..: 2 1 2 1 1 1 2 2 1 2 ...
##  $ Parents.Children.Aboard: Factor w/ 7 levels "0","1","2","3",..: 1 1 1 1 1 3 1 2 1 6 ...
##  $ Fare                   : num  71.28 7.92 53.1 8.05 51.86 ...
```

```
# Convert character variables to numeric
train_x$Age <- as.numeric(train_x$Age)
train_x$Fare <- as.numeric(train_x$Fare)
train_x$Pclass<-as.numeric(train_x$Pclass)
train_x$Sex<-as.numeric(train_x$Sex)
train_x$Siblings.Spouses.Aboard<-as.numeric(train_x$Siblings.Spouses.Aboard)
train_x$Parents.Children.Aboard<-as.numeric(train_x$Parents.Children.Aboard)


# Fit k-NN classifier
knn_model <- knn(train_x, train_x, train_y, k = 3)
summary(knn_model)
```

```
##   0   1
## 383 226
```

```
# Artificial Neural Network (ANN) classifier
ann_model <- nnet(Survived ~ ., data = train, size = 5)
```

```
## # weights:  96
## initial  value 462.304430
## iter  10 value 367.121463
## iter  20 value 306.851246
## iter  30 value 286.135572
## iter  40 value 255.438459
## iter  50 value 243.899092
## iter  60 value 232.328546
## iter  70 value 226.940571
```

```
## iter  80 value 225.407794
## iter  90 value 224.575706
## iter 100 value 223.593753
## final  value 223.593753
## stopped after 100 iterations
```

```r
summary(ann_model)
```

```
## a 17-5-1 network with 96 weights
## options were - entropy fitting
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1
##   34.45   17.62    5.96  -70.30   -6.01  -47.49   -0.72    0.48   -0.18    0.58
## i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1
##   -0.37    0.70    1.28   22.54   -0.46   -0.37   -0.67   13.28
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2
##   -0.84   -0.11   -0.72    0.33   -1.52    0.25   -0.64   -0.34   -0.22   -0.50
## i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##    0.68   -0.20    0.67   -0.38    0.30   -0.19   -0.50   -3.78
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3  i9->h3
##  -30.75  -15.57   21.20  -69.26   10.80  -57.23  -66.27    7.30   43.22   87.80
## i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##    0.53   -1.60  -41.31    1.54   -0.03   -0.31   -0.36    1.20
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4  i9->h4
##   23.89  -21.16  -23.43  -25.01   -0.01   -1.12   -1.94   -1.80   -0.89  -24.21
## i10->h4 i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4
##  -27.14   -0.89   -0.45    0.04  -23.71  -11.34  -17.38    0.00
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5  i9->h5
##   -9.23   56.65  -70.38  -24.60    2.17   31.10    1.10    0.36    1.85   -1.60
## i10->h5 i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5
##   -0.38   13.07   -6.90   -0.23   -0.26   -0.03   -0.31    8.35
##   b->o  h1->o  h2->o  h3->o  h4->o  h5->o
##  58.29   1.07   0.00 -58.99   4.86  -1.88
```

```r
# Naive Bayes classifier
nb_model <- naiveBayes(Survived ~ ., data = train)
summary(nb_model)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    6      -none- list
## levels    2      -none- character
## isnumeric 6      -none- logical
## call      4      -none- call
```

```r
# Support Vector Machine (SVM) classifier
svm_model <- svm(Survived ~ ., data = train)
summary(svm_model)
```

```
##
## Call:
## svm(formula = Survived ~ ., data = train)
##
```

```
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  319
##
##  ( 160 159 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```r
# Decision Tree classifier
tree_model <- rpart(Survived ~ ., data = train)
summary(tree_model)
```

```
## Call:
## rpart(formula = Survived ~ ., data = train)
##   n= 609
##
##           CP nsplit rel error    xerror       xstd
## 1 0.49596774      0 1.0000000 1.0000000 0.04888990
## 2 0.03427419      1 0.5040323 0.5040323 0.04018994
## 3 0.02822581      3 0.4354839 0.4637097 0.03894503
## 4 0.02419355      4 0.4072581 0.4395161 0.03814503
## 5 0.01000000      5 0.3830645 0.4072581 0.03701112
##
## Variable importance
##                     Sex                    Fare Siblings.Spouses.Aboard
##                      46                      16                      10
##                     Age                  Pclass Parents.Children.Aboard
##                      10                      10                       8
##
## Node number 1: 609 observations,    complexity param=0.4959677
##   predicted class=0  expected loss=0.407225  P(node) =1
##     class counts:   361   248
##    probabilities: 0.593 0.407
##   left son=2 (384 obs) right son=3 (225 obs)
##   Primary splits:
##       Sex                     splits as  RL,            improve=95.657250, (0 missing)
##       Fare                    < 10.825   to the left,  improve=28.452000, (0 missing)
##       Pclass                  splits as  RRL,           improve=27.639280, (0 missing)
##       Age                     < 7.5      to the right, improve= 9.625568, (0 missing)
##       Parents.Children.Aboard splits as  LRRRLLL,       improve= 9.411947, (0 missing)
##   Surrogate splits:
##       Parents.Children.Aboard splits as  LRRRLRR,       agree=0.672, adj=0.111, (0 split)
##       Fare                    < 56.7125  to the left,  agree=0.670, adj=0.107, (0 split)
##       Siblings.Spouses.Aboard splits as  LRLRLLL,       agree=0.649, adj=0.049, (0 split)
##       Age                     < 15.5     to the right, agree=0.644, adj=0.036, (0 split)
##
```

```
## Node number 2: 384 observations,    complexity param=0.02822581
##   predicted class=0  expected loss=0.1927083  P(node) =0.6305419
##     class counts:   310    74
##    probabilities: 0.807 0.193
##   left son=4 (363 obs) right son=5 (21 obs)
##   Primary splits:
##       Age                     < 8.5      to the right, improve=9.9805440, (0 missing)
##       Fare                    < 15.1729  to the left,  improve=7.1121290, (0 missing)
##       Pclass                  splits as  RLL,          improve=5.4019580, (0 missing)
##       Parents.Children.Aboard splits as  LRRLLL-,      improve=3.7144730, (0 missing)
##       Siblings.Spouses.Aboard splits as  LRRLLLL,      improve=0.5626304, (0 missing)
##   Surrogate splits:
##       Siblings.Spouses.Aboard splits as  LLLLRLL, agree=0.961, adj=0.286, (0 split)
##
## Node number 3: 225 observations,    complexity param=0.03427419
##   predicted class=1  expected loss=0.2266667  P(node) =0.3694581
##     class counts:    51   174
##    probabilities: 0.227 0.773
##   left son=6 (98 obs) right son=7 (127 obs)
##   Primary splits:
##       Pclass                  splits as  RRL,          improve=17.161860, (0 missing)
##       Fare                    < 48.2     to the left,  improve= 6.866175, (0 missing)
##       Parents.Children.Aboard splits as  RRRRLLL,      improve= 5.743696, (0 missing)
##       Siblings.Spouses.Aboard splits as  RRRLLLL,      improve= 5.171808, (0 missing)
##       Age                     < 14.75    to the left,  improve= 2.164091, (0 missing)
##   Surrogate splits:
##       Fare                    < 19.37915 to the left,  agree=0.769, adj=0.469, (0 split)
##       Age                     < 22.5     to the left,  agree=0.684, adj=0.276, (0 split)
##       Siblings.Spouses.Aboard splits as  RRLLLLL,      agree=0.613, adj=0.112, (0 split)
##       Parents.Children.Aboard splits as  RRLRLLL,      agree=0.604, adj=0.092, (0 split)
##
## Node number 4: 363 observations
##   predicted class=0  expected loss=0.1652893  P(node) =0.5960591
##     class counts:   303    60
##    probabilities: 0.835 0.165
##
## Node number 5: 21 observations,    complexity param=0.02419355
##   predicted class=1  expected loss=0.3333333  P(node) =0.03448276
##     class counts:     7    14
##    probabilities: 0.333 0.667
##   left son=10 (8 obs) right son=11 (13 obs)
##   Primary splits:
##       Siblings.Spouses.Aboard splits as  RRR-LL-,      improve=7.5833330, (0 missing)
##       Pclass                  splits as  RRL,          improve=2.8717950, (0 missing)
##       Fare                    < 29.0625  to the right, improve=2.8717950, (0 missing)
##       Parents.Children.Aboard splits as  -LR----,      improve=0.7619048, (0 missing)
##       Age                     < 3.5      to the right, improve=0.1904762, (0 missing)
##   Surrogate splits:
##       Pclass splits as  RRL,          agree=0.762, adj=0.375, (0 split)
##       Fare   < 29.0625  to the right, agree=0.762, adj=0.375, (0 split)
##
## Node number 6: 98 observations,    complexity param=0.03427419
##   predicted class=1  expected loss=0.4489796  P(node) =0.1609195
##     class counts:    44    54
```

```
##    probabilities: 0.449 0.551
##   left son=12 (21 obs) right son=13 (77 obs)
##   Primary splits:
##       Fare                    < 23.25415 to the right, improve=11.104510, (0 missing)
##       Siblings.Spouses.Aboard splits as  RLRLLLL,      improve= 4.104491, (0 missing)
##       Age                     < 38.5     to the right, improve= 3.835614, (0 missing)
##       Parents.Children.Aboard splits as  RRLRLLL,      improve= 3.012318, (0 missing)
##   Surrogate splits:
##       Siblings.Spouses.Aboard splits as  RRRLLLL,      agree=0.867, adj=0.381, (0 split)
##       Parents.Children.Aboard splits as  RRLRRLL,      agree=0.867, adj=0.381, (0 split)
##       Age                     < 37.5     to the right, agree=0.827, adj=0.190, (0 split)
##
## Node number 7: 127 observations
##   predicted class=1  expected loss=0.05511811  P(node) =0.2085386
##     class counts:     7   120
##    probabilities: 0.055 0.945
##
## Node number 10: 8 observations
##   predicted class=0  expected loss=0.125  P(node) =0.01313629
##     class counts:     7     1
##    probabilities: 0.875 0.125
##
## Node number 11: 13 observations
##   predicted class=1  expected loss=0  P(node) =0.02134647
##     class counts:     0    13
##    probabilities: 0.000 1.000
##
## Node number 12: 21 observations
##   predicted class=0  expected loss=0.0952381  P(node) =0.03448276
##     class counts:    19     2
##    probabilities: 0.905 0.095
##
## Node number 13: 77 observations
##   predicted class=1  expected loss=0.3246753  P(node) =0.1264368
##     class counts:    25    52
##    probabilities: 0.325 0.675
```

```r
# Decision Tree Bagging classifier
bagging_model <- randomForest(Survived ~ ., data = train, mtry = 3, ntree = 10)
summary(bagging_model)
```

```
##                 Length Class  Mode
## call               5   -none- call
## type               1   -none- character
## predicted        609   factor numeric
## err.rate          30   -none- numeric
## confusion          6   -none- numeric
## votes           1218   matrix numeric
## oob.times        609   -none- numeric
## classes            2   -none- character
## importance         6   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
```

```
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y                609   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```r
# Random Forest classifier
rf_model <- randomForest(Survived ~ ., data = train, mtry = 3)
summary(rf_model)
```

```
##                 Length Class  Mode
## call               4   -none- call
## type               1   -none- character
## predicted        609   factor numeric
## err.rate        1500   -none- numeric
## confusion          6   -none- numeric
## votes           1218   matrix numeric
## oob.times        609   -none- numeric
## classes            2   -none- character
## importance         6   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y                609   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```r
# Tuned Random Forest classifier
tuned_rf_model <- randomForest(Survived ~ ., data = train, mtry = 3, nodesize = 5)
summary(tuned_rf_model)
```

```
##                 Length Class  Mode
## call               5   -none- call
## type               1   -none- character
## predicted        609   factor numeric
## err.rate        1500   -none- numeric
## confusion          6   -none- numeric
## votes           1218   matrix numeric
## oob.times        609   -none- numeric
## classes            2   -none- character
## importance         6   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
```

```
## y                 609    factor numeric
## test                0    -none- NULL
## inbag               0    -none- NULL
## terms               3    terms  call
```

```
# Random Forest Boosting classifier
boosting_model <- randomForest(Survived ~ ., data = train, mtry = 3, ntree = 10, method = "adaboost")
summary(boosting_model)
```

```
##                  Length Class  Mode
## call               6    -none- call
## type               1    -none- character
## predicted        609    factor numeric
## err.rate          30    -none- numeric
## confusion          6    -none- numeric
## votes           1218    matrix numeric
## oob.times        609    -none- numeric
## classes            2    -none- character
## importance         6    -none- numeric
## importanceSD       0    -none- NULL
## localImportance    0    -none- NULL
## proximity          0    -none- NULL
## ntree              1    -none- numeric
## mtry               1    -none- numeric
## forest            14    -none- list
## y                609    factor numeric
## test               0    -none- NULL
## inbag              0    -none- NULL
## terms              3    terms  call
```

```
#5. Get confusion matrix and accuracy/misclassification error for all the
library(caret)

# Function to calculate confusion matrix and accuracy
calculate_metrics <- function(model, test_data) {
  # Get predicted class labels
  preds <- predict(model, newdata = test_data[-1])

  # Ensure predicted and reference labels have the same levels
  preds <- factor(preds, levels = levels(test_data$Survived))

  # Create confusion matrix
  cm <- confusionMatrix(preds, test_data$Survived)

  # Extract accuracy from confusion matrix
  accuracy <- cm$overall['Accuracy']

  # Return confusion matrix and accuracy
  return(list(confusion_matrix = cm$table, accuracy = accuracy))
}


# Logistic Regression
```

```r
logistic_results <- calculate_metrics(logit_model, test)
logistic_results
```

```
## $confusion_matrix
##           Reference
## Prediction 0 1
##          0 0 0
##          1 0 0
##
## $accuracy
## Accuracy
##      NaN
```

```r
# K-NN
#knn_results <- calculate_metrics(knn_model, test)

# ANN
ann_results <- calculate_metrics(ann_model, test)
ann_results
```

```
## $confusion_matrix
##           Reference
## Prediction 0 1
##          0 0 0
##          1 3 9
##
## $accuracy
## Accuracy
##     0.75
```

```r
# Naive Bayes
naive_bayes_results <- calculate_metrics(nb_model, test)
naive_bayes_results
```

```
## $confusion_matrix
##           Reference
## Prediction   0   1
##          0 164  42
##          1  20  52
##
## $accuracy
##  Accuracy
## 0.7769784
```

```r
# SVM
svm_results <- calculate_metrics(svm_model, test)
svm_results
```

```
## $confusion_matrix
##           Reference
## Prediction   0   1
```

```
##           0 153  34
##           1  31  60
##
## $accuracy
##  Accuracy
## 0.7661871
```

```r
# Decision Tree
#dt_results <- calculate_metrics(tree_model, test)

# Decision Tree Bagging
bagging_results <- calculate_metrics(bagging_model, test)
bagging_results
```

```
## $confusion_matrix
##            Reference
## Prediction   0   1
##           0 161  31
##           1  23  63
##
## $accuracy
##  Accuracy
## 0.8057554
```

```r
# Random Forest
rf_results <- calculate_metrics(rf_model, test)
rf_results
```

```
## $confusion_matrix
##            Reference
## Prediction   0   1
##           0 166  30
##           1  18  64
##
## $accuracy
##  Accuracy
## 0.8273381
```

```r
# Tuned Random Forest
tuned_rf_results <- calculate_metrics(tuned_rf_model, test)
tuned_rf_results
```

```
## $confusion_matrix
##            Reference
## Prediction   0   1
##           0 166  29
##           1  18  65
##
## $accuracy
##  Accuracy
## 0.8309353
```

```r
# Random Forest Boosting
boosting_results <- calculate_metrics(boosting_model, test)
boosting_results
```

```
## $confusion_matrix
##           Reference
## Prediction   0   1
##          0 163  33
##          1  21  61
##
## $accuracy
##  Accuracy
## 0.8057554
```

```r
#6. Get confusion matrix and accuracy/misclassification error for all the predicted models and interpre
#7. Compare accuracy and misclassification error of predicted models based on "test" data to decide the
#8. Write a reflection on your own word focusing on "what did I learn from this assignment?"
```