Assignment 3 FML

Siddhartha CBS

2023-10-08

##SUMMARY:

we have created a dummy variable “Injury” that classifies MAX_SEV_IR, if injury(Injury = Yes) then Max_Sev_IR equals 1,2 .if injury(Injury = no) then MaX_Sev_IR equals 0.

1. Without any further information is available about the accident the prediction should be INJURY = Yes, because based on the available information the probability of INJURY = yes is greater then INJURY = no, so it is more likely that a newly reported accident will involve an injury than not.

2.Created a subset using first 24 records & and relevant columns(“INJURY”, “WEATHER_R”, “TRAF_CON_R”). Created a pivot table using the subset.

2.1) The 6 possiable combinations are

WEATHER_R = 1 and TRAF_CON_R = 0

WEATHER_R = 1 and TRAF_CON_R = 1

WEATHER_R = 1 and TRAF_CON_R = 2

WEATHER_R = 2 and TRAF_CON_R = 0

WEATHER_R = 2 and TRAF_CON_R = 1

WEATHER_R = 2 and TRAF_CON_R = 2

#Bayes Theorem :

P(A/B) = (P(B/A)P(A))/P(B) where P(A),P(B) are events and P(B) not equal to 0.

conditional probabilities of an injury (INJURY = Yes) & 6 possiable condition

INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0): 0.6666667

INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0

INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2): 0

INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0): 0.1818182

INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1): 0

INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2): 1

conditional probabilities of an injury (INJURY = No) & 6 possiable condition

P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0): 0.3333333

P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1): 1

P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2): 1

P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0): 0.8181818

P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1): 1

P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2): 0

2.2) we have classified the subset by setting a cutoff value of 0.5 i.e probability greater than 0.5 would be classified as “yes” and less than 0.5 as “no”.

2.3)

the naive Bayes conditional probability of an injury(YES) given WEATHER_R = 1 and TRAF_CON_R = 1 is “0”

the naive Bayes conditional probability of an injury(NO) given WEATHER_R = 1 and TRAF_CON_R = 1 is “1”

2.4) Both classification models are giving the same classification result. This suggests that the models are consistently ranking or ordering the observations in the same way. In other words, they are assigning similar relative importance to the data points. The equivalent ranking indicates that both models are assigning equivalent importance to all the factors (attributes or features) used for classification. This implies that the models have a similar understanding of the data, at least in this particular subset.

3.

Data Splitting: dividing our entire dataset into two parts: a training set (typically around 60% of the data) and a validation set (around 40% of the data). This division is essential to ensure that you have data for training your model and data for evaluating its performance.

Validation Set: The validation set is used to assess how well your model has been trained. It serves as a reference point for evaluating the model’s performance when it’s presented with new, unseen data. By comparing the model’s predictions on the validation set to the actual outcomes, you can gauge its effectiveness.

Normalization: Before using the data for training and evaluation, we need to normalize the data. Normalization ensures that all the data features are on the same scale. This step helps improve the accuracy of the model’s predictions by making the data more consistent and removing any biases caused by differing scales or units.

3.1) confusion matrix results:

Miscalculations: 8056

Accuracy : 0.5226

Sensitivity : 0.15957

Specificity : 0.87482

3.2) Error rate = number of misclassified / Total number of Instance

overall error of the validation set is 0.4774209

Questions - Answers

1. if an accident has just been reported and no further information is available, what should the prediction be?

ANS) The prediction should be INJURY = Yes. This is because the majority of accidents in the dataset involve an injury.

2. Select the first 24 records in the dataset with relevent columns & Create a pivot table?

ANS) Created a subset using first 24 records & and relevant columns(“INJURY”, “WEATHER_R”, “TRAF_CON_R”). Created a pivot table using the subset.

2.1) Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes)

ANS) INJURY = Yes

P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0): 0.6666667

P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0

P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2): 0

P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0): 0.1818182

P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1): 0

P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2): 1

2.2) Classify the 24 accidents using these probabilities and a cutoff of 0.5

ANS) probability greater than 0.5 would be classified as “yes” and less than 0.5 as “no”.

2.3) The naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.

ANS)

Probability of INJURY = Yes given WEATHER_R = 1 and TRAF_CON_R = 1 is 0

Probability of INJURY = No given WEATHER_R = 1 and TRAF_CON_R = 1 is 1

2.4) Run a naive Bayes classifier on the 24 records and two predictors.Compare this to the exact Bayes classification. Are the resulting classifications equivalent? Is the ranking (= ordering) of observations equivalent?

ANS) Two predictors classification models are giving the same classification result.Yes the ranking (= ordering) of observations are equivalent.

3.1) what are the results of confusion matrix for the validation data?

ANS) confusion matrix results:

Miscalculations: 8056

Accuracy : 0.5226

Sensitivity : 0.15957

Specificity : 0.87482

3.2) What is the overall error of the validation set?

ANS) overall error of the validation set is 0.4774209

Problem Statement

The file accidentsFull.csv contains information on 42,183 actual automobile accidents in 2001 in the United States that involved one of three levels of injury: NO INJURY, INJURY, or FATALITY. For each accident, additional information is recorded, such as day of week, weather conditions, and road type. A firm might be interested in developing a system for quickly classifying the severity of an accident based on initial reports and associated data in the system (some of which rely on GPS-assisted reporting).

Our goal here is to predict whether an accident just reported will involve an injury (MAX_SEV_IR = 1 or 2) or will not (MAX_SEV_IR = 0). For this purpose, create a dummy variable called INJURY that takes the value “yes” if MAX_SEV_IR = 1 or 2, and otherwise “no.”

#loading the required library

```
library("class")
library("caret")
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library("e1071")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("klaR")
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
#loading the dataset and exploring the data
data.df <- read.csv("C:\\Users\\Siddhartha\\Desktop\\FMA\\accidentsFull.csv")
head(data.df)
```

```
##   HOUR_I_R ALCHL_I ALIGN_I STRATUM_R WRK_ZONE WKDY_I_R INT_HWY LGTCON_I_R
## 1        0       2       2         1        0        1       0          3
## 2        1       2       1         0        0        1       1          3
## 3        1       2       1         0        0        1       0          3
## 4        1       2       1         1        0        0       0          3
## 5        1       1       1         0        0        1       0          3
## 6        1       2       1         1        0        1       0          3
##   MANCOL_I_R PED_ACC_R RELJCT_I_R REL_RWY_R PROFIL_I_R SPD_LIM SUR_COND
## 1          0         0          1         0          1      40        4
## 2          2         0          1         1          1      70        4
## 3          2         0          1         1          1      35        4
## 4          2         0          1         1          1      35        4
## 5          2         0          0         1          1      25        4
## 6          0         0          1         0          1      70        4
##   TRAF_CON_R TRAF_WAY VEH_INVL WEATHER_R INJURY_CRASH NO_INJ_I PRPTYDMG_CRASH
## 1          0        3        1         1            1        1              0
## 2          0        3        2         2            0        0              1
## 3          1        2        2         2            0        0              1
## 4          1        2        2         1            0        0              1
## 5          0        2        3         1            0        0              1
## 6          0        2        1         2            1        1              0
##   FATALITIES MAX_SEV_IR
## 1          0          1
## 2          0          0
## 3          0          0
## 4          0          0
## 5          0          0
## 6          0          1
```

```
dim(data.df)
```

```
## [1] 42183    24
```

```
# Creating a dummy variable INJURY that takes the value "yes" if MAX_SEV_IR = 1 or 2, and "n
o" otherwise
data.df$INJURY <- ifelse(data.df$MAX_SEV_IR %in% c(1, 2), "yes", "no")
t(t(names(data.df)))  #used to display the column names in the dataframe
```

```
##       [,1]
##  [1,] "HOUR_I_R"
##  [2,] "ALCHL_I"
##  [3,] "ALIGN_I"
##  [4,] "STRATUM_R"
##  [5,] "WRK_ZONE"
##  [6,] "WKDY_I_R"
##  [7,] "INT_HWY"
##  [8,] "LGTCON_I_R"
##  [9,] "MANCOL_I_R"
## [10,] "PED_ACC_R"
## [11,] "RELJCT_I_R"
## [12,] "REL_RWY_R"
## [13,] "PROFIL_I_R"
## [14,] "SPD_LIM"
## [15,] "SUR_COND"
## [16,] "TRAF_CON_R"
## [17,] "TRAF_WAY"
## [18,] "VEH_INVL"
## [19,] "WEATHER_R"
## [20,] "INJURY_CRASH"
## [21,] "NO_INJ_I"
## [22,] "PRPTYDMG_CRASH"
## [23,] "FATALITIES"
## [24,] "MAX_SEV_IR"
## [25,] "INJURY"
```

1. Using the information in this dataset, if an accident has just been reported and no further information is available, what should the prediction be? (INJURY = Yes or No?) Why?

```
# Calculate the proportion of accidents with and without injury in the dataset
y <- sum(data.df$INJURY == "yes") #Calculating the number of accidents with injuries
y
```

```
## [1] 21462
```

```
n <- sum(data.df$INJURY == "no")  #Calculating the number of accidents without injuries
n
```

```
## [1] 20721
```

```
# Make the prediction based on the most common outcome
if (y > n) {prediction <- "INJURY = Yes"} else {prediction <- "INJURY = No"}

# Print the prediction
cat("Prediction:", prediction, "\n")
```

```
## Prediction: INJURY = Yes
```

```
#Without any further information about the accident, the prediction should be INJURY = Yes. T
his is because the majority of accidents in the dataset involve an injury. According to the a
ccident data, 50.8% of accidents resulted in an injury (MAX_SEV_IR = 1 or 2), while only 49.
2% resulted in no injury (MAX_SEV_IR = 0). Therefore, based on the available information, it
is more likely that a newly reported accident will involve an injury than not.
```

2. Select the first 24 records in the dataset and look only at the response (INJURY) and the two predictors WEATHER_R and TRAF_CON_R. Create a pivot table that examines INJURY as a function of the two predictors for these 12 records. Use all three variables in the pivot table as rows/columns.

```
# Selecting the first 24 records and relevant columns
subset <- data.df[1:24, c("INJURY", "WEATHER_R", "TRAF_CON_R")]

# Creating the pivot tables
pt1 <- ftable(subset)
pt2 <- ftable(subset[,-1])

# Display the pivot table
print(pt1)
```

```
##                  TRAF_CON_R 0 1 2
## INJURY WEATHER_R
## no     1                   3 1 1
##        2                   9 1 0
## yes    1                   6 0 0
##        2                   2 0 1
```

```
print(pt2)
```

```
##           TRAF_CON_R  0  1  2
## WEATHER_R
## 1                     9  1  1
## 2                    11  1  1
```

2.1 Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.

```
#The 6 possiable combinations are
#WEATHER_R = 1 and TRAF_CON_R = 0
#WEATHER_R = 1 and TRAF_CON_R = 1
#WEATHER_R = 1 and TRAF_CON_R = 2
#WEATHER_R = 2 and TRAF_CON_R = 0
#WEATHER_R = 2 and TRAF_CON_R = 1
#WEATHER_R = 2 and TRAF_CON_R = 2

#INJURY = YES

combination1 = pt1[3,1]/pt2[1,1]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0):", combination1, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0): 0.6666667
```

```
combination2 = pt1[3,2]/pt2[1,2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):", combination2, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0
```

```
combination3 = pt1[3,3]/pt2[1,3]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2):", combination3, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2): 0
```

```
combination4 = pt1[4,1]/pt2[2,1]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0):", combination4, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0): 0.1818182
```

```
combination5 = pt1[4,2]/pt2[2,2]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1):", combination5, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1): 0
```

```
combination6 = pt1[4,3]/pt2[2,3]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2):", combination6, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2): 1
```

```
#These probabilities are based on the data which we have, and used the counts from your pivot tables to calculate the conditional probabilities. These conditional probabilities can be use ful for making Bayesian inferences or predictions based on the available data.


# INJURY = no

box1 = pt1[1,1]/pt2[1,1]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0):", box1, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0): 0.3333333
```

```
box2 = pt1[1,2]/pt2[1,2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", box2, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1): 1
```

```
box3 = pt1[1,3]/pt2[1,3]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2):", box3, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2): 1
```

```
box4 = pt1[2,1]/pt2[2,1]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0):", box4, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0): 0.8181818
```

```
box5 = pt1[2,2]/pt2[2,2]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1):", box5, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1): 1
```

```
box6 = pt1[2,3]/pt2[2,3]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2):", box6, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2): 0
```

```
#These probabilities are based on the data which we have, and used the counts from your pivot tables to calculate the conditional probabilities. These conditional probabilities can be use ful for making Bayesian inferences or predictions based on the available data.
```

2.2 Classify the 24 accidents using these probabilities and a cutoff of 0.5.

```
prob_injury <- rep(0,24)
for(i in 1:24){
  print(c(subset$WEATHER_R[i],subset$TRAF_CON_R[i]))

  if(subset$WEATHER_R[i] == "1" && subset$TRAF_CON_R[i] == "0"){
    prob_injury[i] = combination1}
  else if (subset$WEATHER_R[i] == "1" && subset$TRAF_CON_R[i] == "1"){
    prob_injury[i] = combination2}
  else if (subset$WEATHER_R[i] == "1" && subset$TRAF_CON_R[i] == "2"){
    prob_injury[i] = combination3}
  else if (subset$WEATHER_R[i] == "2" && subset$TRAF_CON_R[i] == "0"){
    prob_injury[i] = combination4}
  else if (subset$WEATHER_R[i] == "2" && subset$TRAF_CON_R[i] == "1"){
    prob_injury[i] = combination5}
  else if(subset$WEATHER_R[i] == "2" && subset$TRAF_CON_R[i] == "2"){
    prob_injury[i] = combination6}}
```

```
## [1] 1 0
## [1] 2 0
## [1] 2 1
## [1] 1 1
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 1 2
## [1] 1 0
## [1] 1 0
## [1] 1 0
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 1 0
## [1] 1 0
## [1] 2 2
## [1] 2 0
```

```
subset$prob_injury = prob_injury
subset$pred.prob  = ifelse(subset$prob_injury>0.5, "yes","no") #setting cutoff of 0.5 for 24 records


head(subset)
```

```
##   INJURY WEATHER_R TRAF_CON_R prob_injury pred.prob
## 1    yes         1          0   0.6666667       yes
## 2     no         2          0   0.1818182        no
## 3     no         2          1   0.0000000        no
## 4     no         1          1   0.0000000        no
## 5     no         1          0   0.6666667       yes
## 6    yes         2          0   0.1818182        no
```

2.3 Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.

```
NY = pt1[3,2]/pt2[1,2]
I = (NY * pt1[3, 2]) / pt2[1, 2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):", NY, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0
```

```
NN = pt1[1,2]/pt2[1,2]
N = (NY * pt1[3, 2]) / pt2[1, 2]
cat("P(YNJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", NN, "\n")
```

```
## P(YNJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1): 1
```

2.4 Run a naive Bayes classifier on the 24 records and two predictors. Check the model output to obtain probabilities and classifications for all 24 records. Compare this to the exact Bayes classification. Are the resulting classifications equivalent? Is the ranking (= ordering) of observations equivalent?

```
new_b <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = subset)

new_buried <- predict(new_b, newdata = subset,type = "raw")
subset$nbpred.prob <- new_buried[,2]


new_c <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
      data = subset, method = "nb")
```

```
## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample21: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBa
yes.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
predict(new_c, newdata = subset[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

```
##  [1] no  no  no  no  no  no  no  no  no  no  no  yes no  no  no  no  no  no  no
## [20] no  no  no  no  no
## Levels: no yes
```

```
predict(new_c, newdata = subset[,c("INJURY", "WEATHER_R", "TRAF_CON_R")],
                                    type = "raw")
```

```
##  [1] no  no  no  no  no  no  no  no  no  no  no  yes no  no  no  no  no  no  no
## [20] no  no  no  no  no
## Levels: no yes
```

3. Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%).

```
##Split Data into 60% training and 40% validation
accident = data.df[c(-24)]

set.seed(123) # Important to ensure that we get the same sample if we rerun the code
accident.index = sample(row.names(accident), 0.6*nrow(accident)[1])
validation.index = setdiff(row.names(accident), accident.index)

acc.data = accident[accident.index,]
valid.data= accident[validation.index,]

dim(acc.data)
```

```
## [1] 25309    24
```

```
dim(valid.data)
```

```
## [1] 16874    24
```

```
norm.values <- preProcess(acc.data[,], method = c("center", "scale"))
acc.norm.data <- predict(norm.values, acc.data[, ])
valid.norm.data <- predict(norm.values, valid.data[, ])

class(acc.norm.data$INJURY)
```

```
## [1] "character"
```

```
acc.norm.data$INJURY <- as.factor(acc.norm.data$INJURY)

class(acc.norm.data$INJURY)
```

```
## [1] "factor"
```

3.1 Run a naive Bayes classifier on the complete training set with the relevant predictors (and INJURY as the response). Note that all predictors are categorical. Show the confusion matrix.

```
nb_model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = acc.norm.data)

predictions <- predict(nb_model, newdata = valid.norm.data)

#Ensure that factor levels in validation dataset match those in training dataset
valid.norm.data$INJURY <- factor(valid.norm.data$INJURY, levels = levels(acc.norm.data$INJUR
Y))

# Show the confusion matrix
confusionMatrix(predictions, valid.norm.data$INJURY)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  1326 1072
##        yes 6984 7492
##
##                Accuracy : 0.5226
##                  95% CI : (0.515, 0.5301)
##     No Information Rate : 0.5075
##     P-Value [Acc > NIR] : 4.725e-05
##
##                   Kappa : 0.0348
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.15957
##             Specificity : 0.87482
##          Pos Pred Value : 0.55296
##          Neg Pred Value : 0.51755
##              Prevalence : 0.49247
##          Detection Rate : 0.07858
##    Detection Prevalence : 0.14211
##       Balanced Accuracy : 0.51720
##
##        'Positive' Class : no
##
```

3.2 What is the overall error of the validation set?

```
# Calculate the overall error rate
error_rate <- 1 - sum(predictions == valid.norm.data$INJURY) / nrow(valid.norm.data)
error_rate
```

```
## [1] 0.4774209
```