# Assignment_3

Pardhu surya sriraj sunkara

#Summary

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?

A.According to the data available there is 50.88 percent chance that the accident could happen if the given condition injury <- yes

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 24 records. Use all three variables in the pivot table as rows/columns.

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 24 records. Use all three variables in the pivot table as rows/columns. 2.1.Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors. A. Predictor combination Probability WEATHER_R = 1 and TRAF_CON_R = 1 0.6666667 WEATHER_R = 2 and TRAF_CON_R = 0 0.1818182 WEATHER_R = 1 and TRAF_CON_R = 1 0.0000000 WEATHER_R = 2 and TRAF_CON_R = 1 0.0000000 WEATHER_R = 1 and TRAF_CON_R = 2 0.0000000 WEATHER_R = 2 and TRAF_CON_R = 2 1.0000000

2.2. Classify the 24 accidents using these probabilities and a cutoff of 0.5. A.Quantitative Prediction: 0.6666667 0.1818182 0.0000000 0.0000000 0.6666667 0.1818182 0.1818182 0.6666667 0.1818182 0.1818182 0.1818182 0.0000000 0.6666667 0.6666667 0.6666667 0.6666667 0.1818182 0.1818182 0.1818182 0.1818182 0.6666667 0.6666667 1.0000000 0.1818182 Qualitative Prediction: “yes” “no” “no” “no” “yes” “no” “no” “yes” “no” “no” “no” “no” “yes” “yes” “yes” “yes” “no” “no” “no” “no” “yes” “yes” “yes” “no”

2.3. Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1. A. The naive Bayes probability of an injury given WEATHER_R=1 and Traf_CON_R=1 is 0

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?

You would need the data and the conditional probabilities linked to the predictors for each class in order to run a naive Bayes classifier on 24 records with two predictors. I’ll give you a rough overview of the procedures you would follow and how to compare the outcomes with accurate Bayes classification because I don’t have access to your data.

A dataset with 24 records and two predictors for each record is required for data preparation. You also need to be aware of the records’ class labels.

Using the training set of data, you would estimate the conditional probabilities for each class when training the Naive Bayes classifier. You presume that the predictors are conditionally independent given the class for a naive Bayes classifier.

3. Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%). 3.1Run 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.

A.
```
Reference
```

Prediction no yes no 3618 4600 yes 3172 5484, Accuracy= 0.534

3.2:-What is the overall error of the validation set? A. The overall error of the validation set is 0.46.

# 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.”

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?
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.

- Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.
- Classify the 24 accidents using these probabilities and a cutoff of 0.5.
- Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.
- 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?

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

- 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.
- What is the overall error of the validation set?

# Summary

# Data Input and Cleaning

Load the required libraries and read the input file

```
library(e1071)
library(caret)
```

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

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

```
accidents <- read.csv("C:/Users/pardh/Downloads/accidentsFull.csv")
accidents$INJURY = ifelse(accidents$MAX_SEV_IR>0,"yes","no")


# Convert variables to factor
for (i in c(1:dim(accidents)[2])){
  accidents[,i] <- as.factor(accidents[,i])
}
head(accidents,n=24)
```

| | HOUR… <fct> | ALCH… <fct> | ALIG… <fct> | STRATU… <fct> | WRK_… <fct> | WKDY… <fct> | INT_… <fct> | LGTCON… <fct> | MANCOL… <fct> |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 2 | 2 | 1 | 0 | 1 | 0 | 3 | 0 |
| 2 | 1 | 2 | 1 | 0 | 0 | 1 | 1 | 3 | 2 |
| 3 | 1 | 2 | 1 | 0 | 0 | 1 | 0 | 3 | 2 |
| 4 | 1 | 2 | 1 | 1 | 0 | 0 | 0 | 3 | 2 |
| 5 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 3 | 2 |
| 6 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 3 | 0 |
| 7 | 1 | 2 | 1 | 0 | 0 | 1 | 1 | 3 | 0 |
| 8 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 3 | 0 |
| 9 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 3 | 0 |
| 10 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 3 | 0 |

1-10 of 24 rows | 1-10 of 26 columns Previous **1** 2 3 Next

# Questions

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.

- Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.
- Classify the 24 accidents using these probabilities and a cutoff of 0.5.
- Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.
- 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?

```
accidents24 <- accidents[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
#head(accidents24)
```

```
dt1 <- ftable(accidents24)
dt2 <- ftable(accidents24[,-1]) # print table only for conditions
dt1
```

```
##                  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
```

```
dt2
```

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

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.

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

```
# Injury = yes
p1 = dt1[3,1] / dt2[1,1] # Injury, Weather=1 and Traf=0
p2 = dt1[4,1] / dt2[2,1] # Injury, Weather=2, Traf=0
p3 = dt1[3,2] / dt2[1,2] # Injury, W=1, T=1
p4 = dt1[4,2] / dt2[2,2] # I, W=2,T=1
p5 = dt1[3,3] / dt2[1,3] # I, W=1,T=2
p6 = dt1[4,3]/ dt2[2,3] #I,W=2,T=2

# Injury = no
n1 = dt1[1,1] / dt2[1,1] # Weather=1 and Traf=0
n2 = dt1[2,1] / dt2[2,1] # Weather=2, Traf=0
n3 = dt1[1,2] / dt2[1,2] # W=1, T=1
n4 = dt1[2,2] / dt2[2,2] # W=2,T=1
n5 = dt1[1,3] / dt2[1,3] # W=1,T=2
n6 = dt1[2,3] / dt2[2,3] # W=2,T=2
print(c(p1,p2,p3,p4,p5,p6))
```

```
## [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.0000000 1.0000000
```

```
print(c(n1,n2,n3,n4,n5,n6))
```

```
## [1] 0.3333333 0.8181818 1.0000000 1.0000000 1.0000000 0.0000000
```

2. Let us now compute

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

```
prob.inj <- rep(0,24)

for (i in 1:24) {
  print(c(accidents24$WEATHER_R[i],accidents24$TRAF_CON_R[i]))
    if (accidents24$WEATHER_R[i] == "1") {
      if (accidents24$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p1
      }
      else if (accidents24$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p3
      }
      else if (accidents24$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p5
      }
    }
    else {
      if (accidents24$TRAF_CON_R[i]=="0"){
        prob.inj[i] = p2
      }
      else if (accidents24$TRAF_CON_R[i]=="1") {
        prob.inj[i] = p4
      }
      else if (accidents24$TRAF_CON_R[i]=="2") {
        prob.inj[i] = p6
      }
    }
  }
```

```
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 1
## Levels: 1 2 0
## [1] 1 1
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 2
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 1 0
## Levels: 1 2 0
## [1] 2 2
## Levels: 1 2 0
## [1] 2 0
## Levels: 1 2 0
```

```
accidents24$prob.inj <- prob.inj
accidents24$prob.inj
```

```
##  [1] 0.6666667 0.1818182 0.0000000 0.0000000 0.6666667 0.1818182 0.1818182
##  [8] 0.6666667 0.1818182 0.1818182 0.1818182 0.0000000 0.6666667 0.6666667
## [15] 0.6666667 0.6666667 0.1818182 0.1818182 0.1818182 0.1818182 0.6666667
## [22] 0.6666667 1.0000000 0.1818182
```

```
accidents24$pred.prob <- ifelse(accidents24$prob.inj>0.5, "yes", "no")
accidents24$pred.prob
```

```
##  [1] "yes" "no"  "no"  "no"  "yes" "no"  "no"  "yes" "no"  "no"  "no"  "no"
## [13] "yes" "yes" "yes" "yes" "no"  "no"  "no"  "no"  "yes" "yes" "yes" "no"
```

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

```
# Probability(Injury=Yes/WEATHER_R=1,TRAF_CON_R=1)

# = [ Probability(W=1/Injury=Yes) * Probability(TRAF_CON_R=1/Injury=Yes) * Probability(Injury=Ye
s)]                                                    /
# [ Probability(W=1/Injury=Yes) * Probability(TRAF_CON_R=1/Injury=Yes) * Probability(Injury=Yes)
+ Probability(WEATHER_R=1/Injury=No) * Probability(TRAF_CON_R=1/Injury=No) * Probability(Injury=
No) ]


# = The result will be "0" since the numerator is equal to zero.
```

2.

- 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?

```
library(klaR)
```

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

```
library(MASS)
```

```
nb <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                 data = accidents24)

nbt <- predict(nb, newdata = accidents24,type = "raw")
accidents24$nbpred.prob <- nbt[,2] # Transfer the "Yes" nb prediction
```

Let us use Caret

```
nb2 <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
      data = accidents24, method = "nb")
```

```
## Warning: model fit failed for Resample01: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample02: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample03: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2, WEATHER_R2
```

```
## Warning: model fit failed for Resample05: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample06: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample08: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample09: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample10: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample11: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample12: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample14: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample15: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample16: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample17: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample18: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample19: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample20: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

```
## Warning: model fit failed for Resample21: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample22: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2, WEATHER_R2
```

```
## Warning: model fit failed for Resample23: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1, TRAF_CON_R2
```

```
## Warning: model fit failed for Resample25: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R1
```

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

```
## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```
predict(nb2, newdata = accidents24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

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

```
predict(nb2, newdata = accidents24[,c("INJURY", "WEATHER_R", "TRAF_CON_R")],
                                    type = "raw")
```

```
##  [1] no no no no no no no no no no no no no no no no no no no no no no 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%). 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.

```
set.seed(2)
train.index <- sample(c(1:dim(accidents)[1]), dim(accidents)[1]*0.6)
train.df <- accidents[train.index,]
valid.df <- accidents[-train.index,]
#defining a variable to be used here
vars <- c("INJURY", "HOUR_I_R",  "ALIGN_I" ,"WRK_ZONE",  "WKDY_I_R",
          "INT_HWY",  "LGTCON_I_R", "PROFIL_I_R", "SPD_LIM", "SUR_COND",
          "TRAF_CON_R",   "TRAF_WAY",   "WEATHER_R")

nbp <- naiveBayes(INJURY~.,data = train.df[,vars])
nbp
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       no      yes
## 0.494014 0.505986
##
## Conditional probabilities:
##      HOUR_I_R
## Y             0         1
##   no  0.5680237 0.4319763
##   yes 0.5776199 0.4223801
##
##      ALIGN_I
## Y             1         2
##   no  0.8714708 0.1285292
##   yes 0.8692019 0.1307981
##
##      WRK_ZONE
## Y              0          1
##   no  0.97608574 0.02391426
##   yes 0.97844760 0.02155240
##
##      WKDY_I_R
## Y             0         1
##   no  0.2192274 0.7807726
##   yes 0.2380915 0.7619085
##
##      INT_HWY
## Y                0            1            9
##   no  0.8493961449 0.1499640086 0.0006398464
##   yes 0.8563954396 0.1429798532 0.0006247072
##
##      LGTCON_I_R
## Y             1         2         3
##   no  0.6951132 0.1222107 0.1826762
##   yes 0.6958457 0.1104951 0.1936592
##
##      PROFIL_I_R
## Y             0         1
##   no  0.7514197 0.2485803
##   yes 0.7633141 0.2366859
##
##      SPD_LIM
## Y                5           10           15           20           25
##   no  0.0001599616 0.0007998080 0.0045589059 0.0079980805 0.1083739902
##   yes 0.0001561768 0.0003904420 0.0046072154 0.0034358894 0.0904263626
##      SPD_LIM
```

```
## Y               30           35           40           45           50
##   no  0.0854194993 0.1933935855 0.0952571383 0.1539630489 0.0399104215
##   yes 0.0891769483 0.2151335312 0.1072934562 0.1534436983 0.0373262533
##      SPD_LIM
## Y               55           60           65           70           75
##   no  0.1618011677 0.0344717268 0.0671838759 0.0403103255 0.0063984644
##   yes 0.1536779635 0.0424019991 0.0637201312 0.0305325629 0.0082773700
##
##      SUR_COND
## Y               1           2           3           4           9
##   no  0.776853555 0.173958250 0.015916180 0.028713109 0.004558906
##   yes 0.815086678 0.153990317 0.010385757 0.015461502 0.005075746
##
##      TRAF_CON_R
## Y             0         1         2
##   no  0.6617612 0.1876350 0.1506039
##   yes 0.6170545 0.2227081 0.1602374
##
##      TRAF_WAY
## Y              1          2          3
##   no  0.57570183 0.37367032 0.05062785
##   yes 0.55825394 0.39879744 0.04294862
##
##      WEATHER_R
## Y             1         2
##   no  0.8420379 0.1579621
##   yes 0.8730283 0.1269717
```

b. What is the overall error of the validation set?

```
ConfusionMatrix = confusionMatrix(valid.df$INJURY, predict(nbp, valid.df[, vars]), positive = "y
es")
print(ConfusionMatrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  3618 4600
##        yes 3172 5484
##
##                Accuracy : 0.5394
##                  95% CI : (0.5319, 0.547)
##     No Information Rate : 0.5976
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0741
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5438
##             Specificity : 0.5328
##          Pos Pred Value : 0.6335
##          Neg Pred Value : 0.4403
##              Prevalence : 0.5976
##          Detection Rate : 0.3250
##    Detection Prevalence : 0.5130
##       Balanced Accuracy : 0.5383
##
##        'Positive' Class : yes
##
```

```
overall_error <- 1 - ConfusionMatrix$overall["Accuracy"]
cat("overall error of the validation set:", overall_error, "\n")
```

```
## overall error of the validation set: 0.4605903
```