Airbag and other influences on accident fatalities

Aditya Mayank Shankar

Description

My project intends to predict the survivability of the passenger in the car depending on various conditions and situations internal to the car and seating of the passenger. The dataset that has been used for making predictions has been obtained by events recorded by the police throughout the USA. To make predictions, it was important to know the nature of the predictor variables and the nature of response variable for making the predictions. Testing the dataset with various techniques for highest accuracy was carried out to select the optimum model for the dataset and implementing the resampling techniques to further better the accuracy was carried out for the dataset.

The dataset has predictor variables which covers majority of factors revolving around an accident scene. These variables are provided in detail and to the point. As a matter of service to society this dataset can be analyzed to find out the chances of survival of a victim when he is bought in emergency to hospital and by knowing some key factors about the accident scene the doctors can be forewarned about the chances of survivability of the victim. The raw dataset can be found at https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/DAAG/nassCDS.csv.

Data Dictionary

Variable Name	Description
Dvcat	Impact Speeds when the accident took place.
Weight	Observation weights, albeit of uncertain accuracy.
Dead	Survivability of Occupant.
Airbag	Existence of Airbag in Car.
Seatbelt	If the Victim had put the seatbelt.
Frontal	If the impact was frontal.
Sex	Sex of the occupant.
aggOfocc	Age of the occupant.
Yearacc	Year of the accident.
Yearveh	Year of the vehicle.
Abcat	If the air-bag was deployed or not.
Occrole	Role of the Occupant.
Deploy	Airbag Deployed or not.

Injseverity	Injury severity the person experienced.
Caseid	The case number created by pasting population sampling unit.

Out of the above mentioned variables, I selected the following variables:

Variable Name	Description
dvcat	Estimated Impact Speeds
airbag	Existence of airbag in a car
seatbelt	Victim was belted or not
frontal	The impact was frontal or not
sex	Sex of the occupant
aggOfocc	Age of the occupant
occRole	Role of the occupant
deploy	Airbag was deployed or not
injseverity	Injury severity the person experienced.

Data Cleaning

1. The dataset was loaded in R and various data cleaning steps where performed.

2. In the injSeverity column, NA values were removed from the dataset using the "na.omit" function which is used to remove the incomplete cases in the dataset.

3. R was considering certain column datatypes as numerical values and "as.factor" function was used to convert the numerical datatypes into categorical datatypes. For Ex: In frontal column, the 0's and 1's in the dataset were taken as numerical values instead of being taken as categorical values. Similar function was used on Airbag.Status and Injury.Severity column.

4. To rename the factor levels, initially a new category was created in the dataset for a specific column with the help of "levels" function. After the creation of a new factor level, it was used to replace the existing factor levels with the desired ones. For Ex: "1-9km/h" was replaced with "1-9" in the dvcat column Similarly, MS Excel had formatting issues and "10-24" was being formatted as "24-Oct" and levels function was used to replace the same with "10-24". "Airbag", "Seatbelt", "Sex", "Occupant.Role" where also cleansed to replace factor levels of desired values.

5. After using the "levels" function to replace the factor levels, "factor" function was used to remove the factor levels which were not needed.

6. To arrange the factor levels in the desired order, "factor" function was used again in the "Vehicle.Speed" column to arrange the impact speeds in the ascending order.

7. %in% function was used to check factor level "5" and to remove it from the "injury.severity" column. After the values were removed, the factor level 6 was renamed with factor level "5".

```r
Dataset <- read.csv(file="D:/Data_Analytics/Airbag and other influences on accident fatalities.csv", header=TRUE, sep=",")

Dataset$X <- NULL
Dataset$weight <- NULL
Dataset$dead <- NULL
Dataset$yearacc <- NULL
Dataset$yearVeh <- NULL
Dataset$abcat <- NULL
Dataset$caseid <- NULL

colnames(Dataset)[1] <- "VehicleSpeed"
colnames(Dataset)[2] <- "Airbag"
colnames(Dataset)[3] <- "Seatbelt"
colnames(Dataset)[4] <- "Frontal"
colnames(Dataset)[5] <- "Sex"
colnames(Dataset)[6] <- "OccupantAge"
colnames(Dataset)[7] <- "OccupantRole"
colnames(Dataset)[8] <- "AirbagStatus"
colnames(Dataset)[9] <- "InjurySeverity"


Dataset$Frontal <- as.factor(Dataset$Frontal)
Dataset$AirbagStatus <- as.factor(Dataset$AirbagStatus)
Dataset$InjurySeverity <- as.factor(Dataset$InjurySeverity)

#Data preprocessing for Vehicle.Speed column
levels(Dataset$VehicleSpeed) <- c(levels(Dataset$VehicleSpeed), "1-9")
Dataset$VehicleSpeed[Dataset$VehicleSpeed=="1-9km/h"]  <- "1-9"

levels(Dataset$VehicleSpeed) <- c(levels(Dataset$VehicleSpeed), "10-24")
Dataset$VehicleSpeed[Dataset$VehicleSpeed=="24-Oct"]  <- "10-24"

Dataset$VehicleSpeed <- factor(Dataset$VehicleSpeed)

Dataset$VehicleSpeed <- factor(Dataset$VehicleSpeed, levels=c("1-9", "10-24", "25-39", "40-54", "55+"))

#Data preprocessing for Airbag column
levels(Dataset$Airbag) <- c(levels(Dataset$Airbag), "Airbag")
Dataset$Airbag[Dataset$Airbag=="airbag"]  <- "Airbag"

levels(Dataset$Airbag) <- c(levels(Dataset$Airbag), "No Airbag")
Dataset$Airbag[Dataset$Airbag=="none"]  <- "No Airbag"

Dataset$Airbag <- factor(Dataset$Airbag)

#Data preprocessing for Seatbelt column
levels(Dataset$Seatbelt) <- c(levels(Dataset$Seatbelt), "Belted")
```

```r
Dataset$Seatbelt[Dataset$Seatbelt=="belted"]  <- "Belted"

levels(Dataset$Seatbelt) <- c(levels(Dataset$Seatbelt), "Not Belted")
Dataset$Seatbelt[Dataset$Seatbelt=="none"]  <- "Not Belted"

Dataset$Seatbelt <- factor(Dataset$Seatbelt)

#Data preprocessing for Sex column
levels(Dataset$Sex) <- c(levels(Dataset$Sex), "M")
Dataset$Sex[Dataset$Sex=="m"]  <- "M"

levels(Dataset$Sex) <- c(levels(Dataset$Sex), "F")
Dataset$Sex[Dataset$Sex=="f"]  <- "F"

Dataset$Sex <- factor(Dataset$Sex)

#Data preprocessing for Occupant.Role column
levels(Dataset$OccupantRole) <- c(levels(Dataset$OccupantRole), "Driver")
Dataset$OccupantRole[Dataset$OccupantRole=="driver"]  <- "Driver"

levels(Dataset$OccupantRole) <- c(levels(Dataset$OccupantRole), "Passanger")
Dataset$OccupantRole[Dataset$OccupantRole=="pass"]  <- "Passanger"

Dataset$OccupantRole <- factor(Dataset$OccupantRole)

#Data preprocessing for Injury.Severity column (Target variable)
Dataset <- Dataset[!Dataset$InjurySeverity %in% c(5), ]
Dataset$InjurySeverity[Dataset$InjurySeverity==6] <- 5
Dataset$InjurySeverity<- factor(Dataset$InjurySeverity)
Dataset$InjurySeverity[Dataset$InjurySeverity==5]  <- 4
Dataset$InjurySeverity <- factor(Dataset$InjurySeverity)

#Command to remove NA's from the dataset
Dataset <- na.omit(Dataset)


summary(Dataset)
```

```
##  VehicleSpeed        Airbag            Seatbelt     Frontal   Sex
##  1-9  :  669   Airbag   :14263   Belted    :18375   0: 9238   M:13816
##  10-24:12699   No Airbag:11668   Not Belted: 7556   1:16693   F:12115
##  25-39: 8129
##  40-54: 2950
##  55+  : 1484
##
##   OccupantAge      OccupantRole   AirbagStatus InjurySeverity
##  Min.   :16.0   Driver   :20441   0:17177      0:6479
##  1st Qu.:22.0   Passanger: 5490   1: 8754      1:5595
##  Median :33.0                                  2:4242
```

```
##  Mean   :37.2                                      3:8495
##  3rd Qu.:48.0                                      4:1120
##  Max.   :97.0
```

Datastructure is as follows

```
str(Dataset)
```

```
## 'data.frame':    25931 obs. of  9 variables:
##  $ VehicleSpeed  : Factor w/ 5 levels "1-9","10-24",..: 3 2 2 3 3 4 5 5 2
2 ...
##  $ Airbag        : Factor w/ 2 levels "Airbag","No Airbag": 2 1 2 1 2 2 2
2 2 2 ...
##  $ Seatbelt      : Factor w/ 2 levels "Belted","Not Belted": 1 1 2 1 1 1 1
2 1 1 ...
##  $ Frontal       : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 1 2 ...
##  $ Sex           : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 1 1 1 2 ...
##  $ OccupantAge   : int  26 72 69 53 32 22 22 32 40 18 ...
##  $ OccupantRole  : Factor w/ 2 levels "Driver","Passanger": 1 1 1 1 1 1 1
1 1 1 ...
##  $ AirbagStatus  : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 1 1 1 1 ...
##  $ InjurySeverity: Factor w/ 5 levels "0","1","2","3",..: 4 2 5 2 4 4 4 5
2 1 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:153] 629 844 1189 1235
1323 1575 2199 2750 2872 3074 ...
##   .. ..- attr(*, "names")= chr [1:153] "639" "863" "1208" "1254" ...
```

Analysis Plan

The response variable of the dataset has been classified in 5 categorical values. Since the output variable is a categorical outcome, the group plans to employ Multiple Logistic regression techniques for predicting the response variable. As the response variable is more than two classes, it would be best suited to use Linear Discriminant Analysis for classification of 5 response variables using the predictor variables. Also, the observations in this dataset can be used to emphasize the use of seatbelts and airbags in the car by studying their influence on the injury severity column.

Since the dataset consists of accidents reported by police, the dataset does not take into consideration non-police-reported car crashes. Also as technology progresses with time, newer car models get introduced in the market and they become safer as well. Since the dataset makes prediction on cars belonging to older technologies, the predictive model does not take into consideration the newer technology cars that will be used by the passengers/drivers in the future.

Training and test data preparation:

Training and test data were split into 70:30 ratio. Our dataset has 25,000+ observations so 70% of the training data would cover most of the cases present in the main dataset and it

will lead to better training for our model. We then tested our model with remaining 30% test data.

```
#Randomizing Dataset

Dataset <- Dataset[sample(1:nrow(Dataset)), ]
head(Dataset)

##       VehicleSpeed    Airbag   Seatbelt Frontal Sex OccupantAge
## 5540         25-39    Airbag     Belted       0   F          55
## 24686        25-39    Airbag     Belted       1   M          18
## 25572        10-24    Airbag Not Belted       1   M          24
## 6828         10-24 No Airbag     Belted       0   F          55
## 19316        25-39 No Airbag     Belted       1   M          54
## 12158        10-24    Airbag     Belted       1   M          45
##       OccupantRole AirbagStatus InjurySeverity
## 5540        Driver            1              3
## 24686       Driver            1              3
## 25572       Driver            1              0
## 6828        Driver            0              3
## 19316       Driver            0              0
## 12158       Driver            1              0

#Splitting the data into training and test data in the ratio 70:30

data_train = Dataset[1:18151,]
data_test = Dataset[18152:25931,]

nrow(data_test)

## [1] 7780

nrow(data_train)

## [1] 18151

summary(Dataset)

##  VehicleSpeed        Airbag            Seatbelt     Frontal   Sex
##  1-9  :  669   Airbag   :14263   Belted    :18375   0: 9238   M:13816
##  10-24:12699   No Airbag:11668   Not Belted: 7556   1:16693   F:12115
##  25-39: 8129
##  40-54: 2950
##  55+  : 1484
##
##   OccupantAge      OccupantRole   AirbagStatus InjurySeverity
##  Min.   :16.0   Driver   :20441   0:17177      0:6479
##  1st Qu.:22.0   Passanger: 5490   1: 8754      1:5595
##  Median :33.0                                  2:4242
##  Mean   :37.2                                  3:8495
```

```
##  3rd Qu.:48.0                                    4:1120
##  Max.   :97.0
```

Model 1: Linear Discriminant Analysis

Since the response variable of our dataset is classified into 6 categories and has multiple class predictor variables, multivariate Linear Discriminant Analysis was the technique used for analyzing the data. To implement LDA function in R, we have used ISLR and MASS libraries.

```
require(ISLR)
```

```
## Loading required package: ISLR
```

```
require(MASS)
```

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

```
fit = lda(InjurySeverity~. , data=data_train)
fit
```

```
## Call:
## lda(InjurySeverity ~ ., data = data_train)
##
## Prior probabilities of groups:
##         0         1         2         3         4
## 0.2499587 0.2137072 0.1645088 0.3285218 0.0433034
##
## Group means:
##   VehicleSpeed10-24 VehicleSpeed25-39 VehicleSpeed40-54 VehicleSpeed55+
## 0         0.6986996         0.2087282        0.02644920     0.004628609
##   AirbagNo Airbag SeatbeltNot Belted  Frontal1      SexF OccupantAge
## 0       0.4253912          0.1452502 0.6528543 0.3828521    35.10007
##   OccupantRolePassanger AirbagStatus1
## 0             0.2018955     0.2603042
##  [ reached getOption("max.print") -- omitted 4 rows ]
##
## Coefficients of linear discriminants:
##                                LD1          LD2         LD3          LD4
## VehicleSpeed10-24      0.615299925 -1.311977968  1.24975239  1.335600859
## VehicleSpeed25-39      1.563041617 -2.264162366  0.66850476  0.988383631
## VehicleSpeed40-54      2.611381571 -1.670767576  0.54733423  0.354563091
## VehicleSpeed55+        3.889023745  1.112368701  1.61061329  2.133004922
## AirbagNo Airbag        0.413801904 -0.411599681 -0.25415474  0.252466225
##  [ reached getOption("max.print") -- omitted 6 rows ]
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4
## 0.8487 0.1146 0.0273 0.0095
```

```
pred = predict(fit, data_test)
pred
```

```
## $class
##  [1] 0 3 1 0 3 1 0 3 0 0 3 0 3 0 0 3 0 3 0 3
##  [ reached getOption("max.print") -- omitted 7760 entries ]
## Levels: 0 1 2 3 4
##
## $posterior
##                  0            1            2          3            4
## 12615 4.854959e-01 0.2365591576 0.1357445921 0.14069131 1.508999e-03
## 2649  1.398009e-01 0.1433430515 0.2366215030 0.47240779 7.826766e-03
## 412   3.221669e-01 0.3558010356 0.1190791119 0.19859067 4.362283e-03
## 10922 4.172056e-01 0.2596129184 0.1750829463 0.14710780 9.907461e-04
##  [ reached getOption("max.print") -- omitted 7776 rows ]
##
## $x
##                 LD1           LD2           LD3           LD4
## 12615 -1.113142e+00  0.7359829228 -0.4333918641  0.4557148909
## 2649   4.562356e-01 -0.7412349462 -1.6363933525 -0.9332991298
## 412   -6.215228e-01  0.6318610584  1.8067543599  0.4705762134
## 10922 -1.118395e+00  0.0501198763 -0.4819036006  1.1509807344
## 1133   3.615903e-01  0.5056114449  0.1573077145 -0.6385036876
##  [ reached getOption("max.print") -- omitted 7775 rows ]
```

```
names(pred)
```

```
## [1] "class"     "posterior" "x"
```

```
pred_class = pred$class
```

```
table(pred_class, data_test$InjurySeverity)
```

```
##
## pred_class    0    1    2    3    4
##          0 1137  678  317  442   16
##          1  243  292  159  228    2
##          2   30   33   19   33    0
##          3  520  692  726 1576  212
##          4   12   21   35  253  104
```

```
mean(pred_class == data_test$InjurySeverity)
```

```
## [1] 0.4020566
```

The Model accuracy was predicted to be 0.4%.

Model 2: Quadratic Discriminant Analysis

We used QDA (quadratic discriminant analysis) which is implemented in R using qda() function and is a part of MASS library. QDA has similar syntax to LDA. The output of QDA is

a mean. The data set was divided into two categories like LDA and predict function was used.

```
fit = qda(InjurySeverity~. , data = data_train)
pred = predict(fit, data_test)
pred
```

```
## $class
##  [1] 0 3 0 0 0 0 0 0 0 0 0 3 0 0 3 0 3 0 3
##  [ reached getOption("max.print") -- omitted 7760 entries ]
## Levels: 0 1 2 3 4
##
## $posterior
##                  0            1            2            3            4
## 12615 7.917066e-01 1.358175e-01 3.169235e-02 4.063199e-02 1.516209e-04
## 2649  1.018199e-01 1.227916e-01 3.444726e-01 4.025066e-01 2.840925e-02
## 412   6.290488e-01 2.986931e-01 3.711013e-02 3.494555e-02 2.024527e-04
## 10922 6.292209e-01 2.005219e-01 1.120290e-01 5.812732e-02 1.009472e-04
##  [ reached getOption("max.print") -- omitted 7776 rows ]
```

```
pred_class = pred$class
```

```
table(pred_class, data_test$InjurySeverity)
```

```
##
## pred_class    0    1    2    3    4
##          0 1582 1162  639  986   36
##          1  136  205  158  224    9
##          2   37   20   47   66    2
##          3  167  290  354  916  146
##          4   20   39   58  340  141
```

```
mean(pred_class == data_test$InjurySeverity)
```

```
## [1] 0.3715938
```

The Model accuracy was predicted to be 0.37%.

Resampling

In the previous deliverable due to inappropriate method of implementation for KNN approach the group was getting an accuracy of 86%. However, after implementing KNN approach with the appropriate procedure as listed in the text book the accuracy dropped down to 40%. As a result, this leaves us with the option of implementing resampling techniques with the LDA model which provides us with the highest accuracy of 42%.

Method 1: Bootstrapping

We used “boot” method from Caret package along with LDA predictive model to implement bootstrapping. The accuracy for bootstrapping with LDA is 41.21%.

```
require(ISLR)
require(MASS)
library(caret)
```

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

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

```
trctrl <- trainControl(method="boot")
set.seed(333)
lda_boot_fit <- train(InjurySeverity~., data=data_train, method="lda",
                      trControl=trctrl,
                      preProcess=c("center", "scale"),
                      tuneLength=10)
lda_boot_fit
```

```
## Linear Discriminant Analysis
##
## 18151 samples
##     8 predictor
##     5 classes: '0', '1', '2', '3', '4'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 18151, 18151, 18151, 18151, 18151, 18151, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.4136267  0.1971207
```

```
test_pred <- predict(lda_boot_fit, newdata=data_test)
test_pred
```

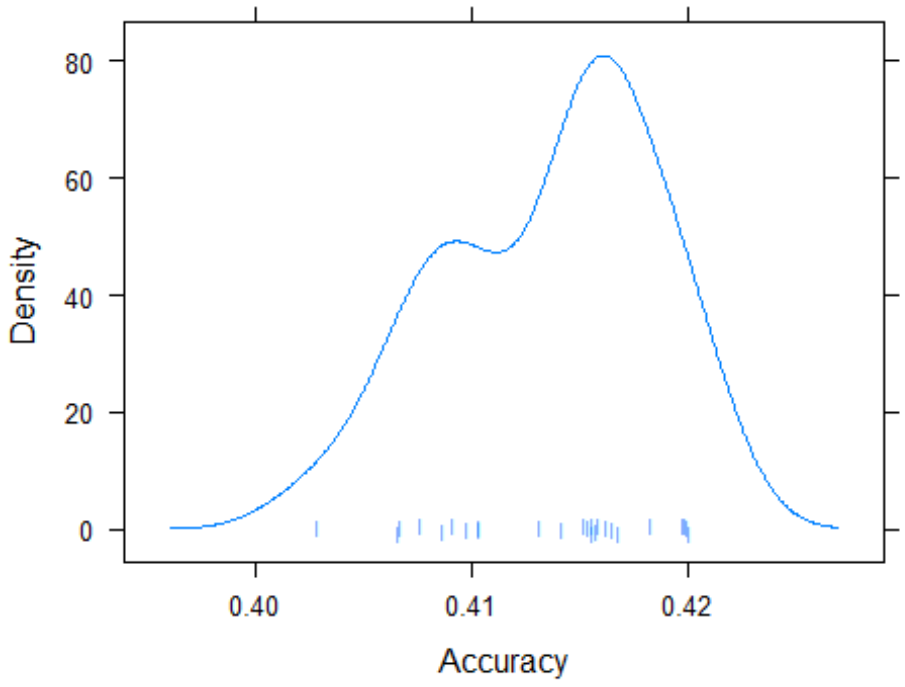
```
##  [1] 0 3 1 0 3 1 0 3 0 0 3 0 3 0 0 3 0 3 0 3
##  [ reached getOption("max.print") -- omitted 7760 entries ]
## Levels: 0 1 2 3 4
```

```
confusionMatrix(test_pred, data_test$InjurySeverity)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1    2    3    4
##          0 1137  678  317  442   16
##          1  243  292  159  228    2
##          2   30   33   19   33    0
##          3  520  692  726 1576  212
##          4   12   21   35  253  104
##
## Overall Statistics
##
##                Accuracy : 0.4021
```

```
##                  95% CI : (0.3911, 0.4131)
##     No Information Rate : 0.3254
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.181
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity            0.5855  0.17016 0.015127   0.6224  0.31138
## Specificity            0.7511  0.89578 0.985285   0.5903  0.95689
## Pos Pred Value         0.4390  0.31602 0.165217   0.4230  0.24471
## Neg Pred Value         0.8449  0.79230 0.838617   0.7642  0.96873
## Prevalence             0.2496  0.22057 0.161440   0.3254  0.04293
## Detection Rate         0.1461  0.03753 0.002442   0.2026  0.01337
## Detection Prevalence   0.3329  0.11877 0.014781   0.4789  0.05463
## Balanced Accuracy      0.6683  0.53297 0.500206   0.6064  0.63413
```

```
densityplot(lda_boot_fit, pch = "|")
```



Method 2: 5-Fold Cross Validation

We used “repeatedcv” method from Caret package along with LDA predictive model to implement bootstrapping. We set number=5 to specify this execution for 5-Fold validation. The accuracy for bootstrapping with LDA is 41.09%.

```r
trctrl <- trainControl(method="repeatedcv", number=5)
set.seed(333)
lda_5fold_fit <- train(InjurySeverity~., data=data_train, method="lda",
                       trControl=trctrl,
                       preProcess=c("center", "scale"),
                       tuneLength=10)
lda_5fold_fit
```

```
## Linear Discriminant Analysis
##
## 18151 samples
##     8 predictor
##     5 classes: '0', '1', '2', '3', '4'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (5 fold, repeated 1 times)
## Summary of sample sizes: 14521, 14521, 14521, 14520, 14521
## Resampling results:
##
##   Accuracy   Kappa
##   0.4130358  0.196888
```

```r
test_pred <- predict(lda_5fold_fit, newdata=data_test)
test_pred
```

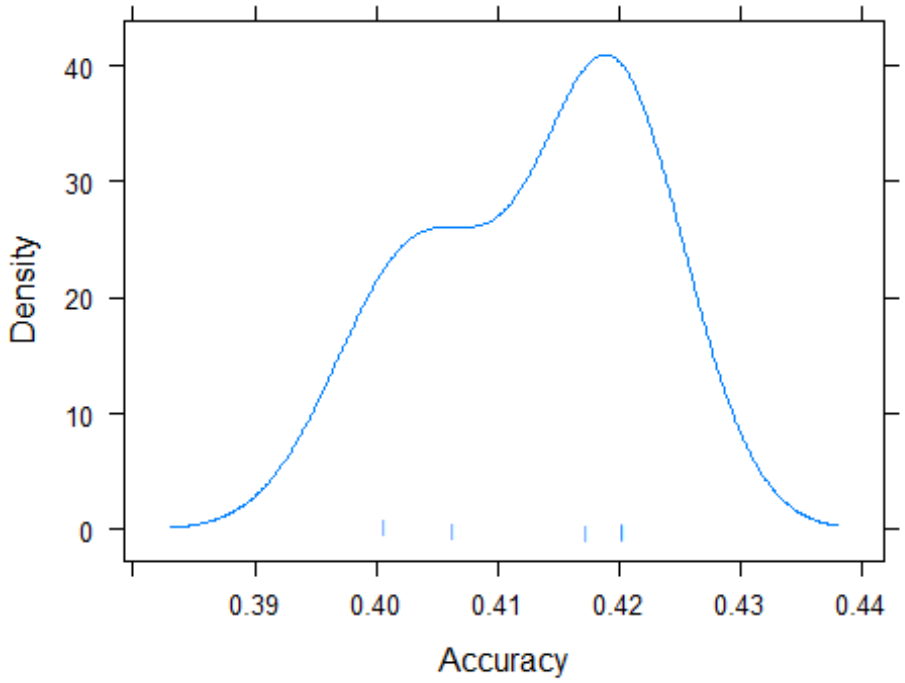
```
##  [1] 0 3 1 0 3 1 0 3 0 0 3 0 3 0 0 3 0 3 0 3
##  [ reached getOption("max.print") -- omitted 7760 entries ]
## Levels: 0 1 2 3 4
```

```r
confusionMatrix(test_pred, data_test$InjurySeverity)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1    2    3    4
##          0 1137  678  317  442   16
##          1  243  292  159  228    2
##          2   30   33   19   33    0
##          3  520  692  726 1576  212
##          4   12   21   35  253  104
##
## Overall Statistics
##
##                Accuracy : 0.4021
##                  95% CI : (0.3911, 0.4131)
##     No Information Rate : 0.3254
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.181
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity            0.5855  0.17016 0.015127   0.6224  0.31138
## Specificity            0.7511  0.89578 0.985285   0.5903  0.95689
## Pos Pred Value         0.4390  0.31602 0.165217   0.4230  0.24471
## Neg Pred Value         0.8449  0.79230 0.838617   0.7642  0.96873
## Prevalence             0.2496  0.22057 0.161440   0.3254  0.04293
## Detection Rate         0.1461  0.03753 0.002442   0.2026  0.01337
## Detection Prevalence   0.3329  0.11877 0.014781   0.4789  0.05463
## Balanced Accuracy      0.6683  0.53297 0.500206   0.6064  0.63413
```

```
densityplot(lda_5fold_fit, pch = "|")
```



Method 3: 10-Fold Cross Validation

We used “repeatedcv” method from Caret package along with LDA predictive model to implement bootstrapping. We set number=10 to specify this execution for 5-Fold validation. The accuracy for bootstrapping with LDA is 41.20%.

```
#LDA with 10-fold

library(caret)
trctrl <- trainControl(method="repeatedcv", number=10)
set.seed(333)
lda_10fold_fit <- train(InjurySeverity~., data=data_train, method="lda",
                        trControl=trctrl,
```

```
                        preProcess=c("center", "scale"),
                        tuneLength=10)


lda_10fold_fit
```

```
## Linear Discriminant Analysis
##
## 18151 samples
##     8 predictor
##     5 classes: '0', '1', '2', '3', '4'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 16336, 16336, 16336, 16337, 16335, 16336, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.4130348  0.1968334
```

```
test_pred <- predict(lda_10fold_fit, newdata=data_test)

test_pred
```

```
##  [1] 0 3 1 0 3 1 0 3 0 0 3 0 3 0 0 3 0 3 0 3
##  [ reached getOption("max.print") -- omitted 7760 entries ]
## Levels: 0 1 2 3 4
```

```
confusionMatrix(test_pred, data_test$InjurySeverity)
```

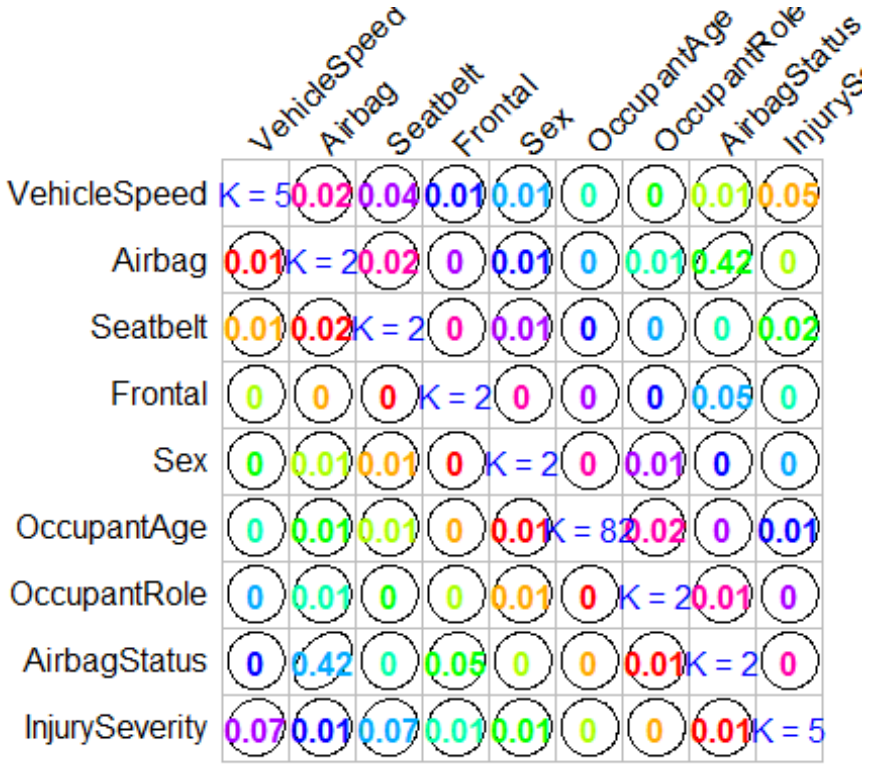
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1    2    3    4
##          0 1137  678  317  442   16
##          1  243  292  159  228    2
##          2   30   33   19   33    0
##          3  520  692  726 1576  212
##          4   12   21   35  253  104
##
## Overall Statistics
##
##                Accuracy : 0.4021
##                  95% CI : (0.3911, 0.4131)
##     No Information Rate : 0.3254
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.181
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
```

```
##
##                      Class: 0 Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity            0.5855  0.17016 0.015127   0.6224  0.31138
## Specificity            0.7511  0.89578 0.985285   0.5903  0.95689
## Pos Pred Value         0.4390  0.31602 0.165217   0.4230  0.24471
## Neg Pred Value         0.8449  0.79230 0.838617   0.7642  0.96873
## Prevalence             0.2496  0.22057 0.161440   0.3254  0.04293
## Detection Rate         0.1461  0.03753 0.002442   0.2026  0.01337
## Detection Prevalence   0.3329  0.11877 0.014781   0.4789  0.05463
## Balanced Accuracy      0.6683  0.53297 0.500206   0.6064  0.63413
```

Correlation Matrix

We have calculated correlation within predictor variables using "Goodman Kruskal" model. Goodman Kruskal model is used to calculate correlation between factor variables. The GoodmanKruskal package includes four functions to compute Goodman and Kruskal's measure and support some simple extensions. These functions are: 1. GKtau is the basic function to compute both the forward association (x,y) and the backward association (y,x) between two categorical vectors x and y. 2. GKtauDataframe computes the Goodman Kruskal association measures between all pairwise combinations of variables in a dataframe, 3. GroupNumeric groups a numeric vector, returning a factor that can be used in association analysis, for reasons discussed in Sections 4 and 5. 4. plot.GKtauMatrix is a plot method for the S3 objects of class GKtauMatrix returned by the GKtauDataframe function.

```
library(GoodmanKruskal)
Cor_Matrix <- GKtauDataframe(Dataset)
plot(Cor_Matrix, backgroundColor = "white", diagColor = "blue", diagSize = 1)
```


VehicleSpeed
Airbag
Seatbelt
Frontal
Sex
OccupantAge
OccupantRole
AirbagStatus
InjurySe
VehicleSpeed K = 5 0.02 0.04 0.01 0.01 0 0 0.01 0.05
Airbag 0.01 K = 2 0.02 0 0.01 0 0.01 0.42 0
Seatbelt 0.01 0.02 K = 2 0 0.01 0 0 0 0.02
Frontal 0 0 0 K = 2 0 0 0 0.05 0
Sex 0 0.01 0.01 0 K = 2 0 0.01 0 0
OccupantAge 0 0.01 0.01 0 0.01 K = 82 0.02 0 0.01
OccupantRole 0 0.01 0 0 0.01 0 K = 2 0.01 0
AirbagStatus 0 0.42 0 0.05 0 0 0.01 K = 2 0
InjurySeverity 0.07 0.01 0.07 0.01 0.01 0 0 0.01 K = 5