Predicting 2000-2014 Fatalities using 1985-1999 data

Caroline Oliver

4/12/2018

Load in the dataset from csv file

```r
library(tree)
library(e1071)
library(readr)

# NOTE: Make sure to replace the path variable with your own path to the file
airline_safety <- read_csv("/Users/carolineoliver/airline-safety.csv")
```

```
## Parsed with column specification:
## cols(
##   airline = col_character(),
##   avail_seat_km_per_week = col_double(),
##   incidents_85_99 = col_integer(),
##   fatal_accidents_85_99 = col_integer(),
##   fatalities_85_99 = col_integer(),
##   incidents_00_14 = col_integer(),
##   fatal_accidents_00_14 = col_integer(),
##   fatalities_00_14 = col_integer()
## )
```

```r
# REPLACE LINE ABOVE WITH YOUR PATH: airline_safety <- read_csv("path_to_csv_file_here")
```

Convert fatality numbers to 1 or 0

```r
i = 0
# Y_N stands for Yes_No
# Yes there were fatalities = 1
# No there were no fatalities = 0
fatalitiesY_N = vector()

for (i in seq(nrow(airline_safety))){
  if(airline_safety$fatalities_00_14[i] == 0){
    fatalitiesY_N[i] = 0
  }
  else{
    fatalitiesY_N[i] = 1
  }
}
```

Create new data frame for prediction

```r
fatal_pred_df = airline_safety[ ,2:5]
View(fatal_pred_df)
```

```
fatal_pred_df$fatalities_00_14_Y_N = fatalitiesY_N
colnames(fatal_pred_df)[5] = "fatalities_00_14_Y_N"
```

Predict all airlines to have fatalities - Error Rate

```
table(fatalitiesY_N)
```

```
## fatalitiesY_N
##  0  1
## 32 24
```

```
32 / 56
```

```
## [1] 0.5714286
```

Predict all airlines to NOT have fatalities - Error Rate

```
table(fatalitiesY_N)
```

```
## fatalitiesY_N
##  0  1
## 32 24
```

```
24 / 56
```

```
## [1] 0.4285714
```

GLM Model

```
glm_model = glm(fatalities_00_14_Y_N ~ ., data=fatal_pred_df)
summary(glm_model)
```

```
##
## Call:
## glm(formula = fatalities_00_14_Y_N ~ ., data = fatal_pred_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6422  -0.3321  -0.2463   0.4424   0.7639
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.710e-01  9.602e-02   2.822  0.00679 **
## avail_seat_km_per_week -1.288e-11  5.051e-11  -0.255  0.79982
## incidents_85_99       -4.487e-03  1.276e-02  -0.352  0.72654
## fatal_accidents_85_99  1.043e-01  6.061e-02   1.721  0.09129 .
## fatalities_85_99      -1.746e-04  5.676e-04  -0.308  0.75971
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2095076)
##
```

```
##     Null deviance: 13.714  on 55  degrees of freedom
## Residual deviance: 10.685  on 51  degrees of freedom
## AIC: 78.156
##
## Number of Fisher Scoring iterations: 2
```

```
glm.pred<-predict(glm_model,fatal_pred_df)
glm.pred
```

```
##         1         2         3         4         5         6         7
## 0.2578808 1.3525458 0.2390978 0.3429782 0.2379951 0.5729358 0.2977018
##         8         9        10        11        12        13        14
## 0.2483816 0.2361222 0.4304810 0.3379518 0.6133547 0.2618870 0.4999882
##        15        16        17        18        19        20        21
## 0.2120952 0.2377337 0.7391418 0.3581469 0.3465427 1.2599390 0.4916140
##        22        23        24        25        26        27        28
## 0.3657911 0.6449188 0.2599784 0.4857638 0.2626191 0.2646273 0.3164080
##        29        30        31        32        33        34        35
## 0.2507984 0.2584409 0.3192262 0.6421703 0.4495765 0.3039037 0.3425194
##        36        37        38        39        40        41        42
## 0.5026854 0.6385794 0.2418101 0.5487693 0.2397582 0.3824926 0.4389779
##        43        44        45        46        47        48        49
## 0.3301793 0.2243098 0.3596858 0.3161439 0.3579715 0.5114794 0.2630145
##        50        51        52        53        54        55        56
## 0.5766397 0.5117883 0.8725969 0.8586392 0.5146083 0.2535558 0.3150579
```

```
average_pred_value = sum(glm.pred)/56

# get 1 or 0 value for prediction
i = 0
predY_N = vector()

for (i in seq(nrow(airline_safety))){
  if(glm.pred[i] > average_pred_value){
    predY_N[i] = 1
  }
  else{
    predY_N[i] = 0
  }
}
```

GLM Confusion matrix

```
table(predict=predY_N,truth=fatalitiesY_N)
```

```
##        truth
## predict  0  1
##       0 26  8
##       1  6 16
```

GLM Error rate

```
(8 + 6) / (26 + 16)
```

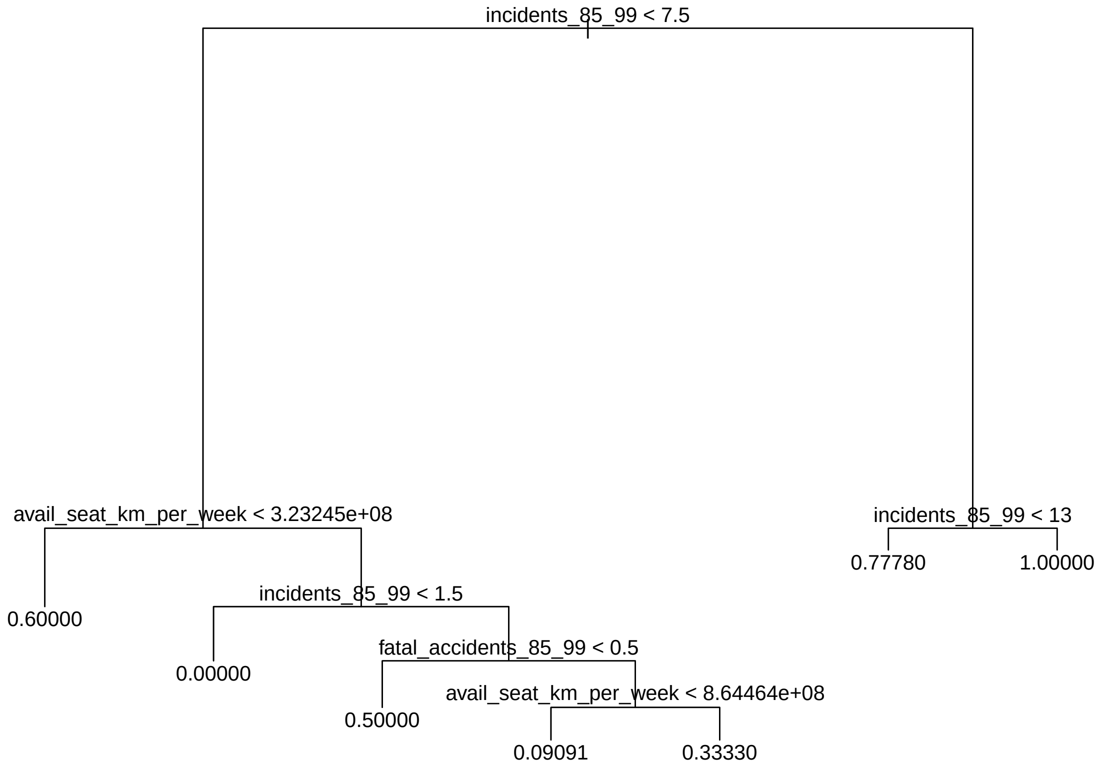
```
## [1] 0.3333333
```

Tree Classification Model

```
fatal.tree<-tree(fatalities_00_14_Y_N ~ ., data=fatal_pred_df)
summary(fatal.tree)
```

```
##
## Regression tree:
## tree(formula = fatalities_00_14_Y_N ~ ., data = fatal_pred_df)
## Variables actually used in tree construction:
## [1] "incidents_85_99"        "avail_seat_km_per_week"
## [3] "fatal_accidents_85_99"
## Number of terminal nodes:  7
## Residual mean deviance:  0.1462 = 7.165 / 49
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.77780 -0.09091  0.00000  0.00000  0.22220  0.90910
```

```
plot(fatal.tree)
text(fatal.tree,pretty=0)
```



```r
fatal_tree.pred<-predict(fatal.tree)
fatal_tree.pred
```

```
##          1          2          3          4          5          6
## 0.60000000 1.00000000 0.50000000 0.09090909 0.50000000 1.00000000
##          7          8          9         10         11         12
## 0.33333333 0.50000000 0.50000000 0.09090909 0.33333333 1.00000000
##         13         14         15         16         17         18
## 0.00000000 0.09090909 0.50000000 0.00000000 0.77777778 0.09090909
##         19         20         21         22         23         24
## 0.09090909 1.00000000 0.77777778 0.00000000 1.00000000 0.00000000
##         25         26         27         28         29         30
## 0.77777778 0.60000000 0.00000000 0.33333333 0.33333333 0.60000000
##         31         32         33         34         35         36
## 0.33333333 0.77777778 0.33333333 0.33333333 0.33333333 0.77777778
##         37         38         39         40         41         42
## 0.09090909 0.00000000 0.60000000 0.50000000 0.09090909 0.33333333
##         43         44         45         46         47         48
## 0.09090909 0.00000000 0.09090909 0.09090909 0.60000000 0.77777778
##         49         50         51         52         53         54
## 0.00000000 0.77777778 0.77777778 1.00000000 1.00000000 0.09090909
##         55         56
## 0.00000000 0.77777778
```

```r
i = 0
tree_predY_N = vector()

for (i in seq(nrow(airline_safety))){
  if(fatal_tree.pred[i] > 0.5){
    tree_predY_N[i] = 1
  }
  else{
    tree_predY_N[i] = 0
  }
}
```

Tree Confusion Matrix

```r
table(tree_predY_N,fatalitiesY_N)
```

```
##              fatalitiesY_N
## tree_predY_N  0  1
##            0 28  7
##            1  4 17
```

Tree Error Rate

```r
(7 + 4) / (28 + 17)
```

```
## [1] 0.2444444
```