NYCFlights: Arrival Delay Logictic Model

Saqib Ali

May 1st, 2017

Logsitic and Inverse Logistic Transformation

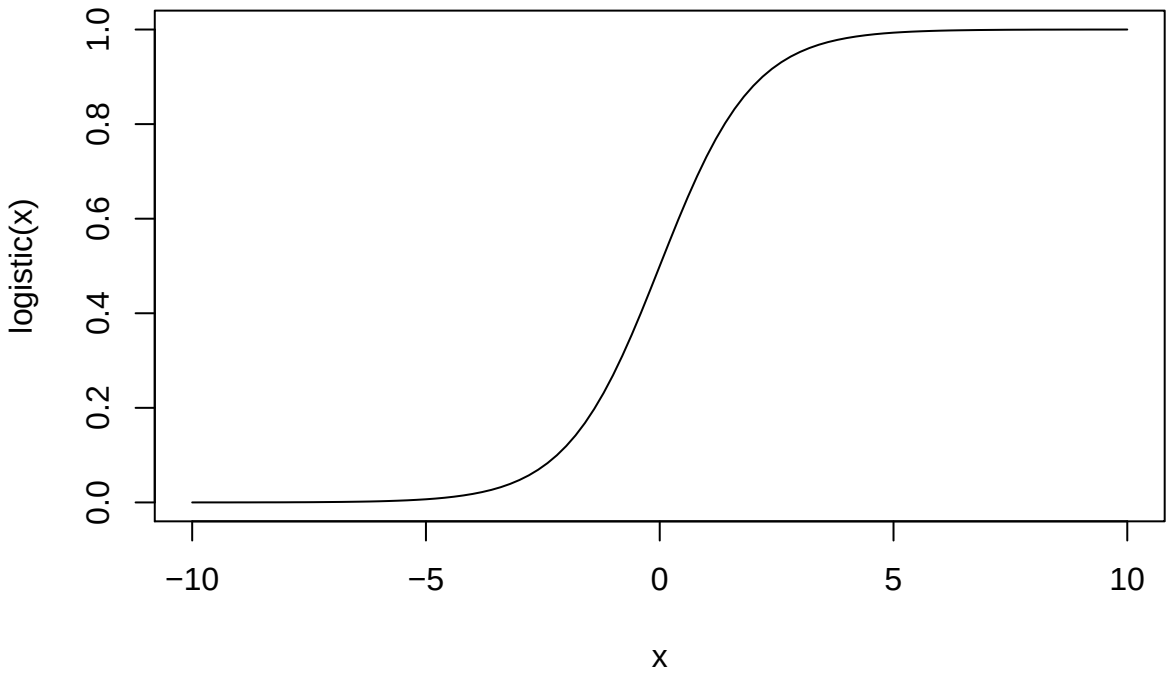
- Write an R function for the logistic function. The function should accept a `numeric` vector with values `[-Inf,Inf]` and produce a numeric vector in the the range `[0,1]`.
- Plot the logistic function from `[-10,10]`
- Write a R function for the inverse logistic function. The function should accept a `numeric` vector with values `[0,1]` and prodcuce a numeric vector in the range `[-Inf,Inf]`
- Plot the Inverse Logistic function from `[0,1]`

Hint: For plotting curves see `?graphics::curve` or `?ggplot2::stat_function`

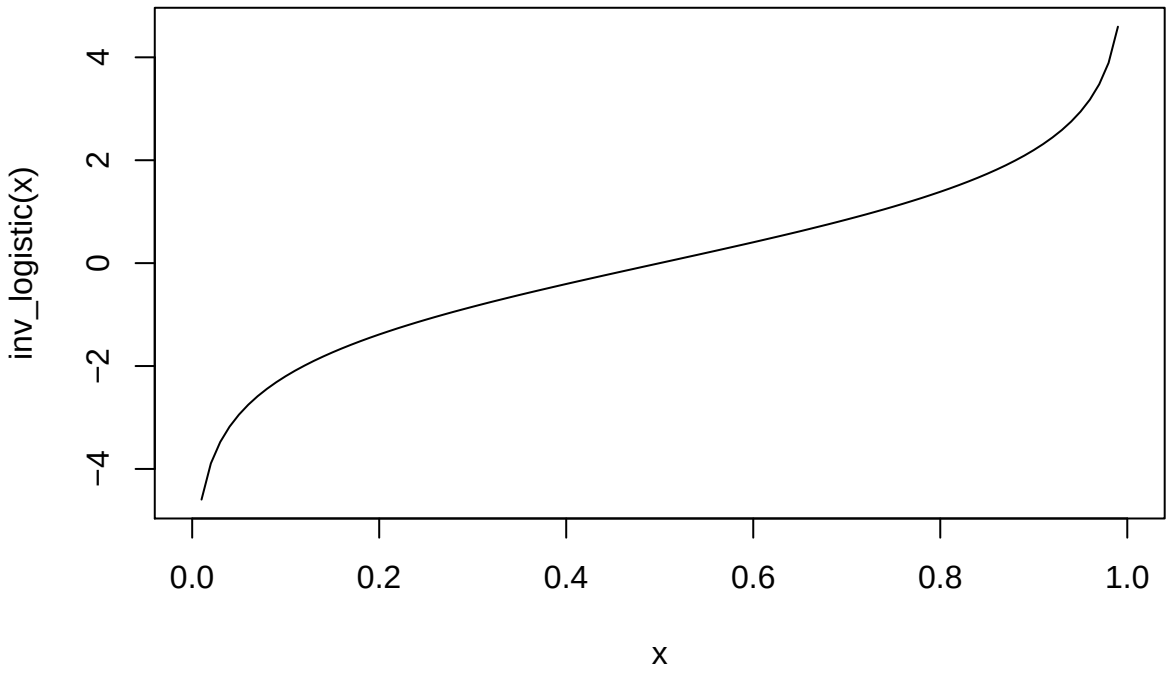
```
logistic <- function(x) {
  1/(1+exp(-x))
}

inv_logistic <- function(x) {
  -log((1-x)/x)
}

x <- -10:10
curve(logistic, -10, 10)
```




```
curve(inv_logistic, 0, 1)
```



NYCFlights Model

Using the rectangular data that you created from the earlier assignment and following theexample from the text and class, create a model for arr_delay >= 22 minutes. Describe/Explain each of the steps and show all work.

KNIT YOUR DOCUMENT AS *HTML* AND SUBMIT IT AND THE `Rmd` file to your repository.

Step 2: Join the datasets and analyze the structure

```
#data.joined <- flights %>% left_join(planes, by = c('tailnum' = 'tailnum')) %>%  left_join(airports, c


YX <- flightsDT
YX %<>% merge( planesDT, all.x = TRUE, by='tailnum', suffixes=c('','.pl') )
YX %<>% merge( weatherDT, all.x = TRUE, by=c('origin','year','month','day','hour'), suffixes=c('','.we'
YX %<>% merge( airportsDT, all.x = TRUE, by.x='origin', by.y='faa', suffixes=c('','.orig') )
YX %<>% merge( airportsDT, all.x = TRUE, by.x='dest', by.y='faa', suffixes=c('','.dest') )


data.joined <- YX
```

Step 3: Add a categorical variable for arr_delay >= 22 minutes. It is called arrival_delayed

```
#data.joined$arrival_delayed <- as.numeric(data.joined$arr_delay >= 22)
```

```
data.joined$arrival_delayed <- ifelse(data.joined$arr_delay >= 22, 1,0)
```

Step 4: Remove entries with NA values for Independant variables that we want to use in the model

```
data.joined <- data.joined  %>%
  filter(!is.na(dep_delay))   %>%
  filter(!is.na(dest))   %>%
  filter(!is.na(origin))   %>%
  filter(!is.na(year))   %>%
  filter(!is.na(month))   %>%
  filter(!is.na(day))   %>%
  filter(!is.na(hour))   %>%
  filter(!is.na(tailnum))   %>%
  filter(!is.na(sched_dep_time))   %>%
  filter(!is.na(sched_arr_time))   %>%
  filter(!is.na(flight))   %>%
  filter(!is.na(distance))   %>%
  filter(!is.na(year.pl))   %>%
  filter(!is.na(minute))   %>%
  filter(!is.na(year.pl))   %>%
  filter(!is.na(type))   %>%
  filter(!is.na(manufacturer))   %>%
  filter(!is.na(model))   %>%
  filter(!is.na(engines))   %>%
  filter(!is.na(seats))   %>%
  filter(!is.na(engine))   %>%
  filter(!is.na(temp))   %>%
  filter(!is.na(dewp))   %>%
  filter(!is.na(humid))   %>%
  filter(!is.na(wind_dir))   %>%
  filter(!is.na(wind_speed))   %>%
  filter(!is.na(wind_gust))   %>%
  filter(!is.na(precip))   %>%
  filter(!is.na(pressure))   %>%
  filter(!is.na(visib))   %>%
  filter(!is.na(name))   %>%
  filter(!is.na(lat))   %>%
  filter(!is.na(lon))   %>%
  filter(!is.na(tz))   %>%
  filter(!is.na(name.dest))   %>%
  filter(!is.na(lat.dest))   %>%
  filter(!is.na(lon.dest))   %>%
  filter(!is.na(alt.dest))   %>%
  filter(!is.na(tz.dest))
```

Step 5: Get sample training data (50% of the entire dataset)

```
data.joined.training <- sample_frac(data.joined, .75)
data.joined.testing <- sample_frac(data.joined, .5)
```

Step 6: Generate a Logistic Model

```
logit.fit <- glm(arrival_delayed ~  dep_delay + dest + origin + year + month + day + hour + sched_dep_t
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logit.fit)
```

```
##
## Call:
## glm(formula = arrival_delayed ~ dep_delay + dest + origin + year +
##     month + day + hour + sched_dep_time + sched_arr_time + carrier +
##     distance + year.pl + type + engines + seats + engine + temp +
##     dewp + humid + wind_dir + wind_speed + wind_gust + precip +
##     pressure + visib + lat + lon + alt + tz + lat.dest + lon.dest +
##     alt.dest + tz.dest, family = binomial, data = data.joined.training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2807  -0.3182  -0.2420  -0.1794   3.7884
##
## Coefficients: (10 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  4.168e+01  7.191e+00   5.797 6.77e-09 ***
## dep_delay                    1.063e-01  6.396e-04 166.173  < 2e-16 ***
## destACK                     -1.940e+00  4.279e+00  -0.453 0.650312
## destALB                     -2.811e+00  4.391e+00  -0.640 0.522136
## destANC                      6.882e-01  5.314e+00   0.130 0.896949
## destATL                     -7.568e-01  2.808e+00  -0.269 0.787563
## destAUS                     -7.898e-02  8.616e-01  -0.092 0.926961
## destAVL                     -7.971e-01  3.247e+00  -0.245 0.806097
## destBDL                     -2.998e+00  4.462e+00  -0.672 0.501629
## destBGR                     -2.256e+00  3.818e+00  -0.591 0.554466
## destBHM                     -2.275e+00  2.556e+00  -0.890 0.373420
## destBNA                     -1.122e+00  2.801e+00  -0.401 0.688753
## destBOS                     -2.026e+00  4.275e+00  -0.474 0.635581
## destBTV                     -2.133e+00  4.084e+00  -0.522 0.601498
## destBUF                     -1.881e+00  4.007e+00  -0.469 0.638735
## destBUR                      1.528e+00  1.714e+00   0.892 0.372507
## destBWI                     -1.718e+00  4.316e+00  -0.398 0.690651
## destBZN                      1.346e+00  7.545e-01   1.783 0.074514 .
## destCAE                      6.045e-01  3.205e+00   0.189 0.850398
## destCAK                     -1.595e+00  3.765e+00  -0.424 0.671842
## destCHO                     -2.779e+00  4.091e+00  -0.679 0.496935
## destCHS                     -1.263e+00  3.121e+00  -0.405 0.685632
## destCLE                     -1.598e+00  3.701e+00  -0.432 0.665957
## destCLT                     -1.250e+00  3.369e+00  -0.371 0.710655
## destCMH                     -1.428e+00  3.549e+00  -0.402 0.687496
## destCRW                     -1.101e+00  3.931e+00  -0.280 0.779344
## destCVG                     -1.091e+00  3.269e+00  -0.334 0.738645
## destDAY                     -1.591e+00  3.366e+00  -0.473 0.636433
## destDCA                     -1.615e+00  4.229e+00  -0.382 0.702586
```

```
## destDEN                    4.130e-01  6.177e-01   0.669 0.503746
## destDFW                   -1.371e-01  1.195e+00  -0.115 0.908616
## destDSM                   -1.003e+00  2.117e+00  -0.474 0.635439
## destDTW                   -1.611e+00  3.480e+00  -0.463 0.643358
## destEGE                   -9.388e-01  6.377e-01  -1.472 0.140940
## destEYW                    1.396e+00  1.834e+00   0.761 0.446566
## destFLL                   -3.650e-01  1.993e+00  -0.183 0.854705
## destGRR                   -1.075e+00  3.181e+00  -0.338 0.735386
## destGSO                   -1.463e+00  3.594e+00  -0.407 0.684002
## destGSP                   -1.142e+00  3.208e+00  -0.356 0.721876
## destHDN                    5.576e-01  1.069e+00   0.521 0.602029
## destHNL                    3.698e+00  8.261e+00   0.448 0.654391
## destHOU                    5.447e-02  1.089e+00   0.050 0.960118
## destIAD                   -1.776e+00  4.195e+00  -0.423 0.672063
## destIAH                    1.529e-01  1.121e+00   0.136 0.891555
## destILM                   -2.871e+00  3.545e+00  -0.810 0.417996
## destIND                   -1.079e+00  3.072e+00  -0.351 0.725419
## destJAC                    7.046e-01  7.897e-01   0.892 0.372288
## destJAX                   -8.957e-01  2.622e+00  -0.342 0.732666
## destLAS                    8.771e-01  1.139e+00   0.770 0.441130
## destLAX                    1.434e+00  1.719e+00   0.835 0.403892
## destLEX                   -8.711e+00  1.970e+02  -0.044 0.964730
## destLGB                    7.755e-01  1.712e+00   0.453 0.650565
## destMCI                   -3.978e-01  1.910e+00  -0.208 0.834995
## destMCO                   -8.791e-01  2.322e+00  -0.379 0.704929
## destMDW                   -1.055e+00  2.899e+00  -0.364 0.715946
## destMEM                   -7.289e-01  2.287e+00  -0.319 0.749959
## destMHT                   -2.271e+00  4.222e+00  -0.538 0.590577
## destMIA                   -4.937e-01  1.942e+00  -0.254 0.799300
## destMKE                   -8.500e-01  2.864e+00  -0.297 0.766602
## destMSN                   -1.206e+00  2.680e+00  -0.450 0.652726
## destMSP                   -7.733e-01  2.128e+00  -0.363 0.716262
## destMSY                   -5.808e-01  1.714e+00  -0.339 0.734721
## destMTJ                   -1.088e+00  1.660e+00  -0.656 0.512059
## destMVY                   -1.631e+00  4.348e+00  -0.375 0.707613
## destMYR                   -1.540e+00  3.386e+00  -0.455 0.649318
## destOAK                    1.033e+00  2.007e+00   0.515 0.606750
## destOKC                    3.690e-01  1.329e+00   0.278 0.781303
## destOMA                   -2.965e-01  1.808e+00  -0.164 0.869723
## destORD                   -9.117e-01  2.877e+00  -0.317 0.751371
## destORF                   -1.738e+00  4.024e+00  -0.432 0.665832
## destPBI                   -4.708e-01  2.100e+00  -0.224 0.822563
## destPDX                    1.234e+00  1.669e+00   0.740 0.459526
## destPHL                   -1.212e+00  4.541e+00  -0.267 0.789565
## destPHX                    5.570e-01  9.038e-01   0.616 0.537733
## destPIT                   -1.592e+00  3.915e+00  -0.407 0.684368
## destPSP                   -6.650e+00  5.341e+01  -0.125 0.900904
## destPVD                   -2.273e+00  4.346e+00  -0.523 0.601059
## destPWM                   -2.016e+00  4.058e+00  -0.497 0.619350
## destRDU                   -1.586e+00  3.672e+00  -0.432 0.665799
## destRIC                   -1.554e+00  4.033e+00  -0.385 0.700031
## destROC                   -1.876e+00  4.106e+00  -0.457 0.647663
## destRSW                   -7.181e-01  1.984e+00  -0.362 0.717452
## destSAN                    1.460e+00  1.644e+00   0.888 0.374304
```

```
## destSAT                       -2.976e-01  7.313e-01  -0.407 0.684096
## destSAV                       -1.108e+00  2.913e+00  -0.380 0.703665
## destSBN                       -3.449e+00  3.549e+00  -0.972 0.331042
## destSDF                       -1.005e+00  3.082e+00  -0.326 0.744283
## destSEA                        1.219e+00  1.584e+00   0.769 0.441617
## destSFO                        1.570e+00  2.004e+00   0.783 0.433334
## destSJC                        1.226e+00  1.984e+00   0.618 0.536544
## destSLC                        4.331e-01  5.287e-01   0.819 0.412669
## destSMF                        1.468e+00  1.861e+00   0.789 0.430301
## destSNA                        7.298e-01  1.667e+00   0.438 0.661505
## destSRQ                       -6.388e-01  2.075e+00  -0.308 0.758219
## destSTL                       -8.108e-01  2.479e+00  -0.327 0.743622
## destSYR                       -2.478e+00  4.249e+00  -0.583 0.559805
## destTPA                       -5.202e-01  2.164e+00  -0.240 0.810074
## destTUL                       -9.353e-02  1.616e+00  -0.058 0.953853
## destTVC                       -1.802e+00  3.176e+00  -0.567 0.570496
## destTYS                       -1.198e+00  3.115e+00  -0.385 0.700553
## destXNA                        1.338e-01  1.823e+00   0.073 0.941497
## originJFK                     -1.268e-01  5.062e-02  -2.505 0.012254 *
## originLGA                      6.237e-02  4.935e-02   1.264 0.206316
## year                                  NA         NA      NA       NA
## month                          2.985e-03  3.047e-03   0.980 0.327261
## day                            1.237e-03  1.157e-03   1.069 0.285018
## hour                          -5.981e-02  5.430e-02  -1.101 0.270713
## sched_dep_time                 8.355e-04  5.426e-04   1.540 0.123655
## sched_arr_time                 7.378e-05  3.270e-05   2.256 0.024067 *
## carrierAA                     -1.110e-01  9.827e-02  -1.130 0.258472
## carrierAS                     -3.030e-01  2.830e-01  -1.071 0.284261
## carrierB6                      5.148e-01  5.837e-02   8.819  < 2e-16 ***
## carrierDL                     -1.162e-01  6.666e-02  -1.744 0.081236 .
## carrierEV                      2.661e-01  6.207e-02   4.287 1.81e-05 ***
## carrierF9                      7.313e-01  1.833e-01   3.991 6.59e-05 ***
## carrierFL                      5.338e-01  1.200e-01   4.449 8.63e-06 ***
## carrierHA                      8.354e-01  5.334e-01   1.566 0.117275
## carrierMQ                     -2.873e-02  2.168e-01  -0.133 0.894579
## carrierOO                      1.150e+00  8.650e-01   1.329 0.183695
## carrierUA                     -2.050e-01  7.310e-02  -2.804 0.005047 **
## carrierUS                      5.907e-01  8.146e-02   7.252 4.11e-13 ***
## carrierVX                     -1.585e-01  1.031e-01  -1.538 0.123985
## carrierWN                     -3.416e-01  9.134e-02  -3.740 0.000184 ***
## carrierYV                      1.043e-01  2.536e-01   0.411 0.680939
## distance                      -1.278e-03  2.620e-03  -0.488 0.625865
## year.pl                       -1.379e-02  2.482e-03  -5.557 2.74e-08 ***
## typeFixed wing single engine   6.400e-01  6.250e-01   1.024 0.305832
## typeRotorcraft                 4.706e-02  9.927e-01   0.047 0.962188
## engines                        1.254e-01  2.143e-01   0.585 0.558673
## seats                         -3.955e-04  2.429e-04  -1.628 0.103506
## engineReciprocating           -1.710e-01  7.548e-01  -0.227 0.820745
## engineTurbo-fan                2.953e-01  9.105e-01   0.324 0.745642
## engineTurbo-jet                4.407e-01  9.107e-01   0.484 0.628428
## engineTurbo-prop              -5.695e-01  1.395e+00  -0.408 0.682991
## engineTurbo-shaft                     NA         NA      NA       NA
## temp                          -1.254e-02  6.299e-03  -1.990 0.046551 *
## dewp                           8.740e-03  6.766e-03   1.292 0.196454
```

```
## humid                         5.853e-04  3.357e-03   0.174 0.861613
## wind_dir                     -8.808e-04  1.038e-04  -8.490  < 2e-16 ***
## wind_speed                    5.823e+02  2.912e+02   1.999 0.045558 *
## wind_gust                    -5.060e+02  2.531e+02  -1.999 0.045559 *
## precip                        3.300e+00  5.813e-01   5.677 1.37e-08 ***
## pressure                     -1.495e-02  1.509e-03  -9.906  < 2e-16 ***
## visib                        -5.877e-02  7.568e-03  -7.765 8.13e-15 ***
## lat                                 NA         NA      NA       NA
## lon                                 NA         NA      NA       NA
## alt                                 NA         NA      NA       NA
## tz                                  NA         NA      NA       NA
## lat.dest                            NA         NA      NA       NA
## lon.dest                            NA         NA      NA       NA
## alt.dest                            NA         NA      NA       NA
## tz.dest                             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 166836  on 175238  degrees of freedom
## Residual deviance:  73288  on 175097  degrees of freedom
##   (570 observations deleted due to missingness)
## AIC: 73572
##
## Number of Fisher Scoring iterations: 10
```

Question:

Is this a good model? (Write your answer here.)

To answer this let's create a Confusion Matrix. Based on the Confusion Matrix results, this is a fairly good model, with high accuracy, sensitivity and specivity.

```
#prob = predict(logit.fit, data.joined.sample, type="response")
prob = plogis(predict(logit.fit, data.joined.testing))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
prob<-ifelse(prob> 0.5,1,0)
data.joined.testing$prob = prob


confusionMatrix(data.joined.testing$arrival_delayed, prob)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 93334  2016
##          1  6563 14910
##
##                Accuracy : 0.9266
```

```
##                  95% CI : (0.9251, 0.9281)
##     No Information Rate : 0.8551
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7334
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9343
##             Specificity : 0.8809
##          Pos Pred Value : 0.9789
##          Neg Pred Value : 0.6944
##              Prevalence : 0.8551
##          Detection Rate : 0.7989
##    Detection Prevalence : 0.8162
##       Balanced Accuracy : 0.9076
##
##        'Positive' Class : 0
##
```

PART B:

Your model should be good at explaining tardiness. Now, assume that your job is to predict arrival delays a month in advance. You can no longer use all the features in your model. Retrain your model using only features that will be *known* only a month in advance of the departure time. Show all steps as above.

```
logit.fit <- glm(arrival_delayed ~  dest + origin + month + day + hour + sched_dep_time + sched_arr_time
```

Let's see how good this model is using a Confusion Matrix. Based on the outcome of the Confusion Matrix, this doesn't look a good Model as the Specificity is very low.

```
#prob = predict(logit.fit, data.joined.sample, type="response")
prob = plogis(predict(logit.fit, data.joined.testing))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
prob<-ifelse(prob> 0.5,1,0)
data.joined.testing$prob = prob


confusionMatrix(data.joined.testing$arrival_delayed, prob)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 95320    30
##          1 21446    27
##
##                Accuracy : 0.8162
##                  95% CI : (0.8139, 0.8184)
##     No Information Rate : 0.9995
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0015
```

```
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.816334
##             Specificity : 0.473684
##          Pos Pred Value : 0.999685
##          Neg Pred Value : 0.001257
##              Prevalence : 0.999512
##          Detection Rate : 0.815935
##    Detection Prevalence : 0.816192
##       Balanced Accuracy : 0.645009
##
##        'Positive' Class : 0
##
```