

sumo

Loading

We load our favorite packages, set a random number seed and define helper functions. Then load the data and split it.

Training

```
bin <- glm(result ~ ., tr, family = 'binomial')
nn <- neuralnet(result ~ ., normalize(tr), hidden = 32)

bin.ans <- predict.glm(bin, ev)
nn.ans <- predict(nn, ev)
```

Evaluation

```
results <- tibble(bin.ans) %>%
  cbind(nn.ans) %>% cbind(ev$result)

results$bin.ans = sapply(results$bin.ans,
  function(x) {if (x > 0) {
    TRUE
  } else {
    FALSE}})

results$nn.ans = sapply(results$nn.ans,
  function(x) {if (x >= 0.5) {
    TRUE
  } else {
    FALSE}})

colnames(results) <- c('glm', 'nn', 'truth')

confusionMatrix(as.factor(results$glm), as.factor(results$truth))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE      5      2
```

```

##      TRUE      3      4
##
##              Accuracy : 0.6429
##              95% CI : (0.3514, 0.8724)
##      No Information Rate : 0.5714
##      P-Value [Acc > NIR] : 0.399
##
##              Kappa : 0.2857
##
##      McNemar's Test P-Value : 1.000
##
##              Sensitivity : 0.6250
##              Specificity : 0.6667
##              Pos Pred Value : 0.7143
##              Neg Pred Value : 0.5714
##              Prevalence : 0.5714
##              Detection Rate : 0.3571
##      Detection Prevalence : 0.5000
##              Balanced Accuracy : 0.6458
##
##      'Positive' Class : FALSE
##

```

```
confusionMatrix(as.factor(results$nn), as.factor(results$truth))
```

```

## Confusion Matrix and Statistics
##
##              Reference
## Prediction FALSE TRUE
##      FALSE      7      6
##      TRUE       1      0
##
##              Accuracy : 0.5
##              95% CI : (0.2304, 0.7696)
##      No Information Rate : 0.5714
##      P-Value [Acc > NIR] : 0.7918
##
##              Kappa : -0.1395
##
##      McNemar's Test P-Value : 0.1306
##
##              Sensitivity : 0.8750
##              Specificity : 0.0000
##              Pos Pred Value : 0.5385
##              Neg Pred Value : 0.0000
##              Prevalence : 0.5714
##              Detection Rate : 0.5000
##      Detection Prevalence : 0.9286
##              Balanced Accuracy : 0.4375
##
##      'Positive' Class : FALSE
##

```

```
if (bin.ans > 0) {
  print('GLM says left.')
} else {print('GLM says right.')}
```

```
## Warning in if (bin.ans > 0) {: the condition has length > 1 and only the first
## element will be used
```

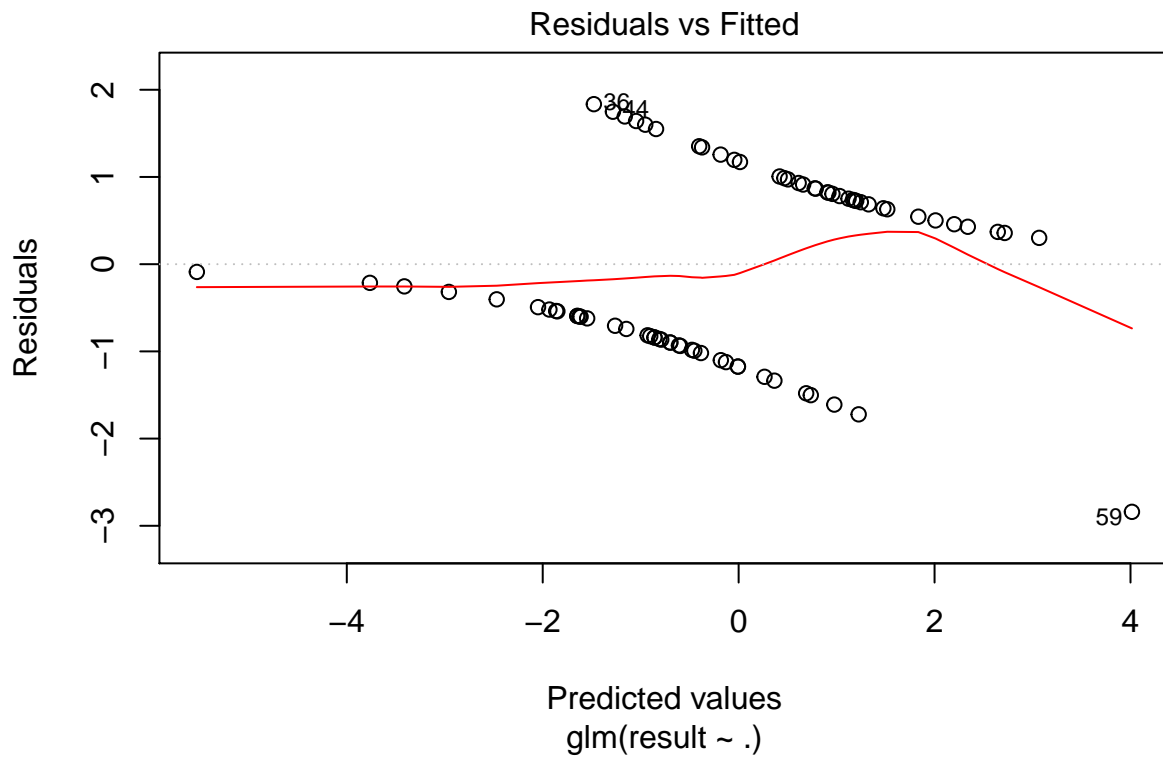
```
## [1] "GLM says right."
```

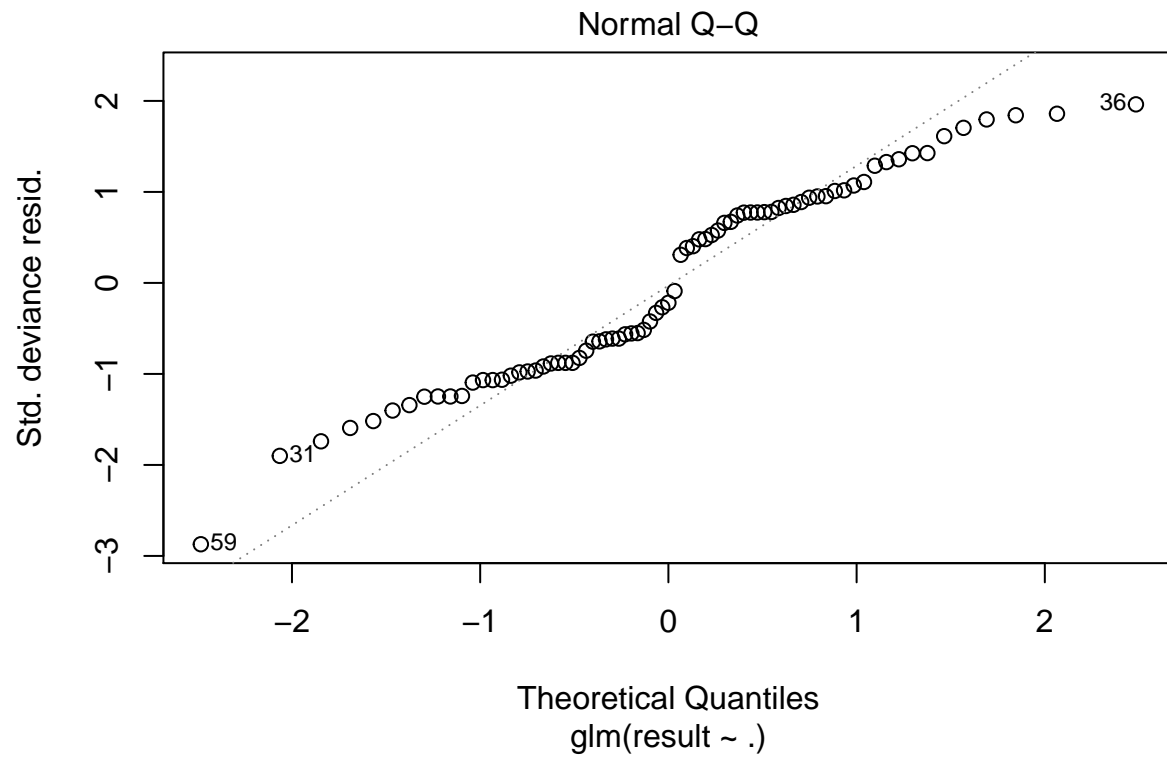
```
if (nn.ans >= 0.5) {
  print('NN says left.')
} else {print('NN says right.')}
```

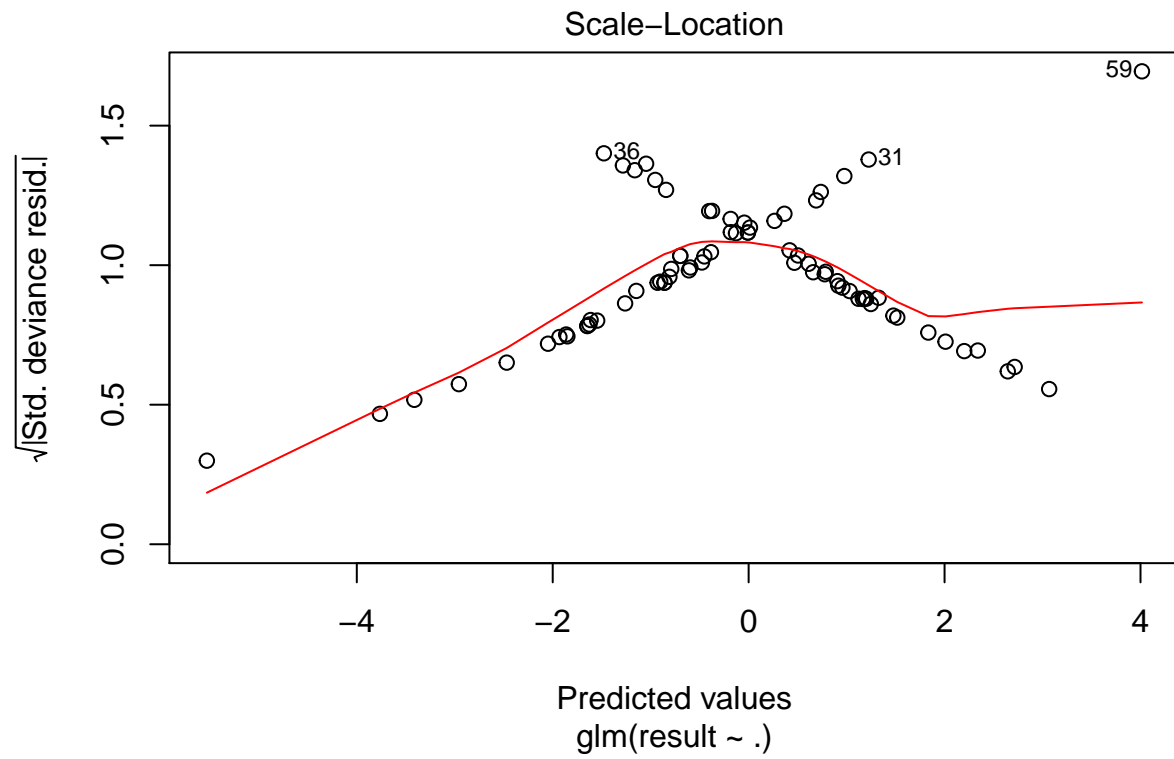
```
## Warning in if (nn.ans >= 0.5) {: the condition has length > 1 and only the first
## element will be used
```

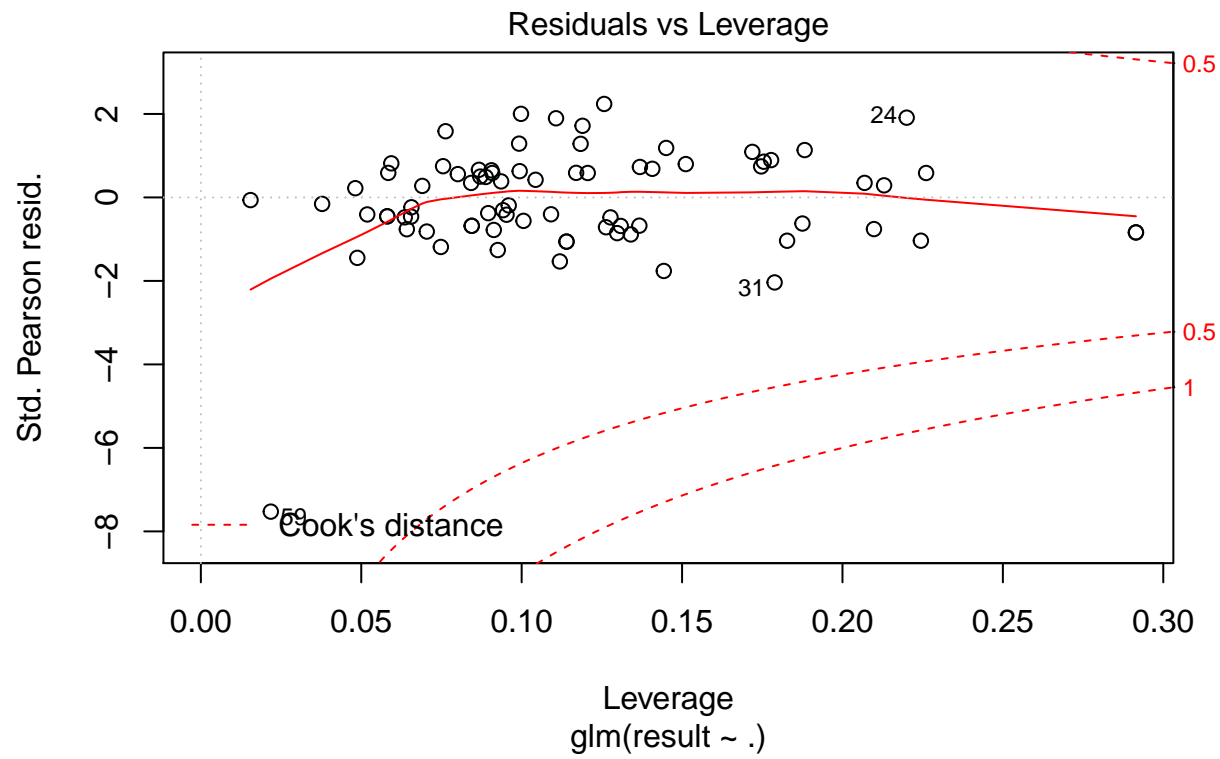
```
## [1] "NN says right."
```

```
plot(bin)
```









```
plot(nn)
bin.ans
```

```
##          1          2          3          4          5          6
## -0.42306619  1.24730357 -1.19357536  1.20004141  0.07127859  0.13615763
##          7          8          9         10         11         12
## -0.50529109  0.33625829 -1.25990949 -0.95386939  1.78797885 -0.21140848
##          13         14
##  0.04901999 -0.89915586
```

```
nn.ans
```

```
##          [,1]
## [1,]  0.01225422
## [2,] -0.15458215
## [3,]  0.21716041
## [4,] -0.82544796
## [5,] -1.25512187
## [6,] -1.25504767
## [7,]  0.21716041
## [8,] -0.15454394
## [9,]  0.21715787
## [10,] -0.50783439
## [11,]  0.55794264
## [12,]  0.20455634
```

```
## [13,] -1.25510773  
## [14,] -0.62638846
```