Abalone

Shiva Sankar Modala

2023-02-09

```
## Installing the necessary packages for the problem ##

#install.packages('readr')     ## abalone.data is a large dataset. So, I used readr package in handling that data
#install.packages('knitr')     ## To convert the r script into the markdown ans later for presentation, Knit is used for documentation.
#install.packages('stringr')   ## It provides a cohesive set of functions designed to work with strings easily
#install.packages('caret')     ## To use machine learning models, I used caret package to fit our model
#install.packages('corrplot')  ## With the corrplot, I can provide the correlation matrix for our data.
#install.packages('pROC')      ## For the ROC curves and analysis.

## These are the libraries that I used for the abalone data.
library(readr)
library(knitr)
library(stringr)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(corrplot)

## corrplot 0.92 loaded

library(pROC)

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

# Reading the abalone data as the csv format
data_abalone= read.csv('https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data',header = FALSE,sep = ",",stringsAsFactors = TRUE)
```

```r
# Remove the Infants in the observations by keeping the Male/Female classes
infant_remove = subset(data_abalone,V1!='I')
infant_remove$V1 = factor(infant_remove$V1)
set.seed(1)

# With the help of createDataPartition() in the caret package, we split the
data into 80% and 20%.
partition_data = createDataPartition(infant_remove$V1,p=0.2,list=FALSE)

# Dividing the test data and train data by separating the columns.
# test data has the infant data with the data part
test_data = infant_remove[partition_data,]
# Train data is without that data part
train_data = infant_remove[-partition_data,]

# Fit a logistic regression using all feature variables using the generalized
linear models
# I used glm to apply that model to the data
log_regression = glm(V1~V2+V3+V4+V5+V6+V7+V8+V9,data=train_data,family =
binomial)

# Summary for the above logistic regression
summary(log_regression)

##
## Call:
## glm(formula = V1 ~ V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9, family =
binomial,
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8773  -1.1995   0.8723   1.1165   1.5184
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.858543   0.520622   5.491 4.00e-08 ***
## V2          -0.629068   2.292027  -0.274   0.7837
## V3          -6.633627   2.709837  -2.448   0.0144 *
## V4          -3.732314   2.249421  -1.659   0.0971 .
## V5          -0.745165   0.854026  -0.873   0.3829
## V6           4.055672   1.027483   3.947 7.91e-05 ***
## V7          -1.041244   1.442155  -0.722   0.4703
## V8           1.368821   1.299135   1.054   0.2920
## V9           0.001171   0.018057   0.065   0.9483
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3128.9  on 2266  degrees of freedom
```

```
## Residual deviance: 3064.5  on 2258  degrees of freedom
## AIC: 3082.5
##
## Number of Fisher Scoring iterations: 4
```

```
# Coefficient for the above logistic regression
coef(log_regression)
```

```
##  (Intercept)           V2           V3           V4           V5
V6
##  2.858543140 -0.629067560 -6.633626737 -3.732313567 -0.745165230
4.055671602
##           V7           V8           V9
## -1.041243887  1.368820970  0.001170528
```

```
cat("\n The null hypothesis can be avoided for the variables for which the
predictions have a lower p-value")
```

```
##
##  The null hypothesis can be avoided for the variables for which the
predictions have a lower p-value
```

```
cat("\n We can tell from the output that V3 and V6 are the important
predictors.")
```

```
##
##  We can tell from the output that V3 and V6 are the important predictors.
```

```
# Now we have to present the confidence intervals for the logistic regression
confint(log_regression)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %      97.5 %
## (Intercept)   1.85352256  3.89549890
## V2           -5.12145416  3.86968345
## V3          -11.96790846 -1.33704996
## V4           -8.56672822 -0.04177129
## V5           -2.44078468  0.91942538
## V6            2.05920944  6.09531362
## V7           -3.86758608  1.79335994
## V8           -1.17091097  3.93439914
## V9           -0.03424511  0.03659261
```

```
cat("\n Confidence interval does not contain 0 for V6 but it does for V3. V6
has 95% chance that +  predictor V6 falls between range 2.05920944 &
6.09531362 and we can reject the null hypothesis.")
```

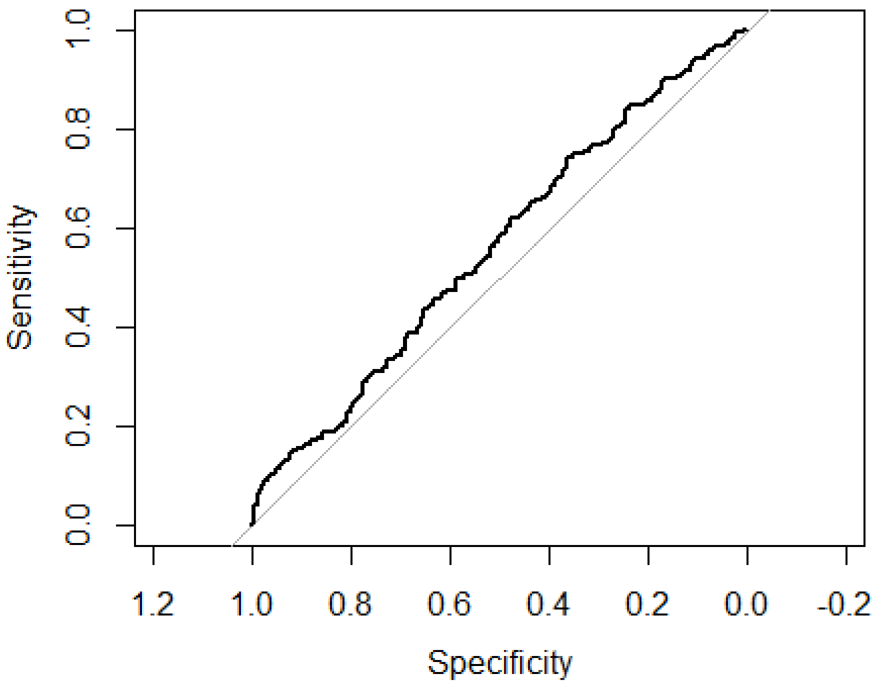
```
##
##  Confidence interval does not contain 0 for V6 but it does for V3. V6 has
95% chance that +  predictor V6 falls between range 2.05920944 & 6.09531362
and we can reject the null hypothesis.
```

```r
# The type as response provides the predicted probabilities
predic1= predict(log_regression,test_data,type="response")

# Create a new variable for the male and female and this can help us in
making the confusion matrix
predic = ifelse(predic1>=0.5,'M','F')

# Confusion matrix for predictor for the test dataset.
confusionMatrix(as.factor(predic),as.factor(test_data$V1))
```

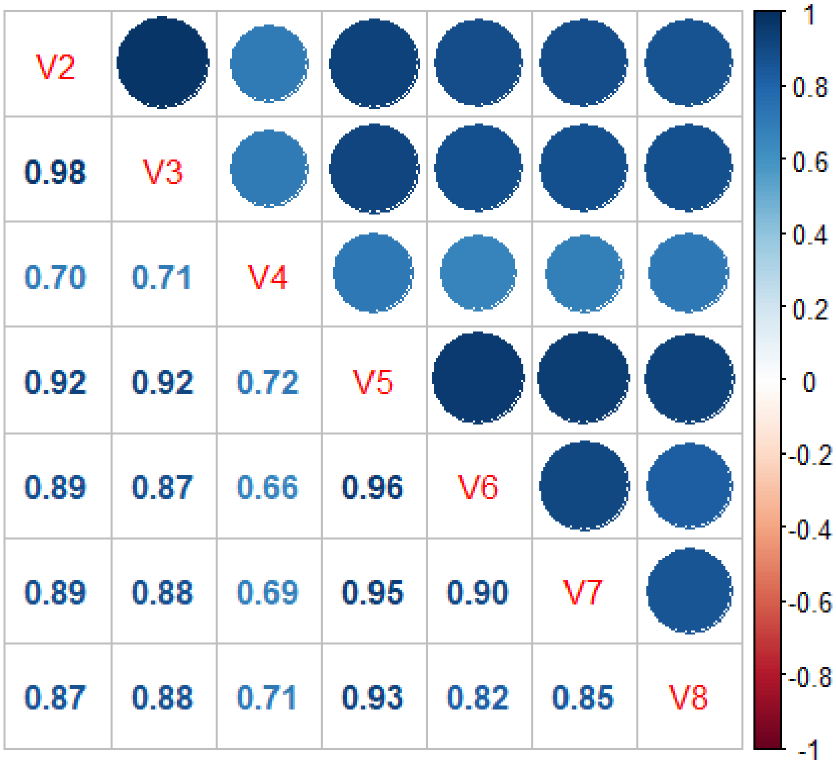
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   F   M
##          F  97  85
##          M 165 221
##
##                Accuracy : 0.5599
##                  95% CI : (0.5179, 0.6012)
##     No Information Rate : 0.5387
##     P-Value [Acc > NIR] : 0.1665
##
##                   Kappa : 0.0945
##
##  Mcnemar's Test P-Value : 5.841e-07
##
##             Sensitivity : 0.3702
##             Specificity : 0.7222
##          Pos Pred Value : 0.5330
##          Neg Pred Value : 0.5725
##              Prevalence : 0.4613
##          Detection Rate : 0.1708
##    Detection Prevalence : 0.3204
##       Balanced Accuracy : 0.5462
##
##        'Positive' Class : F
##
```

```r
# plotting the ROC curve for the predictor
plot(roc(test_data$V1,predic1))
```

```
## Setting levels: control = F, case = M

## Setting direction: controls < cases
```




```
cat("\n As we can see ROC curve is better for our model")

##
##  As we can see ROC curve is better for our model

cat("hence it will predict better than selecting random value")

## hence it will predict better than selecting random value

cat("Accuracy of the model is 0.5599")

## Accuracy of the model is 0.5599

# plotting the mixed Correlation plot for the model
corrplot.mixed(cor(infant_remove[,2:8]))
```




```
# Conclusion
cat("\n Given that the above plot donesn't explain much, the strong correlation between all the variables demonstrates the classifier's poor performance")

##
##  Given that the above plot donesn't explain much, the strong correlation between all the variables demonstrates the classifier's poor performance

cat("\n A good model has uncorrelated variables.")

##
##  A good model has uncorrelated variables.
```