**POM681 Business Analytics & Data Mining**

Prof. Rai

**Template for Assignment 6: Classification and Prediction**

Student Name:

* Pranav Vinod

**Overview:** In this analysis you will develop logistic regression model based on the data set provided to predict whether or not the specimens are genuine.

**Data Set Information:** Data (A6DATA.txt) were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.

**Attribute Information:**

V1: variance of Wavelet Transformed image (continuous)

V2: skewness of Wavelet Transformed image (continuous)

V3: kurtosis of Wavelet Transformed image (continuous)

V4: entropy of image (continuous)

V5: class (0-forged, 1-genuine)

1. (5 points) Read the A6DATA.csv data file into RStudio. Run set.seed(222) for partitioning of the dataset into training (50%) and testing (50%). Report on the number of forged and genuine banknote-like specimens in the training and testing data.

# Read the data  
data <- read.csv("/Users/pranavvinod/downloads/A6DATA.txt", header = F)  
data$V5 <- as.factor(data$V5)  
  
summary(data) # 762 forged banknotes and 610 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0421 Min. :-13.773 Min. :-5.2861 Min. :-8.5482   
## 1st Qu.:-1.7730 1st Qu.: -1.708 1st Qu.:-1.5750 1st Qu.:-2.4135   
## Median : 0.4962 Median : 2.320 Median : 0.6166 Median :-0.5867   
## Mean : 0.4337 Mean : 1.922 Mean : 1.3976 Mean :-1.1917   
## 3rd Qu.: 2.8215 3rd Qu.: 6.815 3rd Qu.: 3.1793 3rd Qu.: 0.3948   
## Max. : 6.8248 Max. : 12.952 Max. :17.9274 Max. : 2.4495   
## V5   
## 0:762   
## 1:610   
##   
##   
##   
##

set.seed(222)  
  
# Partitioning the data  
indices <- sample(1:nrow(data), size = nrow(data)\*0.5)  
train <- data[indices,]  
test <- data[-indices,]  
  
# Data description  
summary(train) # 381 forged banknotes and 305 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0364 Min. :-13.773 Min. :-5.2133 Min. :-7.7853   
## 1st Qu.:-1.8180 1st Qu.: -2.502 1st Qu.:-1.2489 1st Qu.:-2.0381   
## Median : 0.3345 Median : 1.883 Median : 0.7679 Median :-0.5168   
## Mean : 0.3549 Mean : 1.673 Mean : 1.5917 Mean :-1.0993   
## 3rd Qu.: 2.6627 3rd Qu.: 6.623 3rd Qu.: 3.3064 3rd Qu.: 0.4385   
## Max. : 6.5633 Max. : 12.952 Max. :17.9274 Max. : 2.4495   
## V5   
## 0:381   
## 1:305   
##   
##   
##   
##

summary(test) # 381 forged banknotes and 305 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0421 Min. :-13.678 Min. :-5.2861 Min. :-8.5482   
## 1st Qu.:-1.7448 1st Qu.: -1.017 1st Qu.:-1.8191 1st Qu.:-2.6168   
## Median : 0.5404 Median : 2.872 Median : 0.3065 Median :-0.7160   
## Mean : 0.5126 Mean : 2.172 Mean : 1.2035 Mean :-1.2840   
## 3rd Qu.: 2.9713 3rd Qu.: 6.937 3rd Qu.: 3.1330 3rd Qu.: 0.3408   
## Max. : 6.8248 Max. : 12.378 Max. :17.6772 Max. : 2.1625   
## V5   
## 0:381   
## 1:305

**There are 381 forged and 305 genuine banknotes in both the training and testing dataset.**

1. (20 points) Develop a logistic regression model using the training data. Provide final logistic regression model (with only significant variables), equation for calculating probability that specimen is genuine, confusion matrix for both training & testing data, misclassification error for both training & testing data, and discuss performance of the model.

# Create model for 50:50 split  
m1 <- glm(formula = V5 ~., data = train, family = 'binomial')

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

summary(m1)

##   
## Call:  
## glm(formula = V5 ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.21019 0.00000 0.00000 0.00008 2.34895   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.5831 3.7680 2.543 0.01098 \*   
## V1 -9.2371 3.3643 -2.746 0.00604 \*\*  
## V2 -3.7759 1.2996 -2.905 0.00367 \*\*  
## V3 -5.2246 1.8005 -2.902 0.00371 \*\*  
## V4 0.9020 0.8068 1.118 0.26358   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 942.561 on 685 degrees of freedom  
## Residual deviance: 21.758 on 681 degrees of freedom  
## AIC: 31.758  
##   
## Number of Fisher Scoring iterations: 13

# we can see that V4 is not a significant variable for predictions, so we drop it  
m1.best <- glm(formula = V5 ~.-V4, data = train, family = 'binomial')

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

summary(m1.best)

##   
## Call:  
## glm(formula = V5 ~ . - V4, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.34409 0.00000 0.00000 0.00013 2.03675   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.094 2.613 3.098 0.001951 \*\*   
## V1 -8.616 2.688 -3.205 0.001349 \*\*   
## V2 -3.948 1.174 -3.364 0.000770 \*\*\*  
## V3 -5.217 1.585 -3.291 0.000999 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 942.561 on 685 degrees of freedom  
## Residual deviance: 23.373 on 682 degrees of freedom  
## AIC: 31.373  
##   
## Number of Fisher Scoring iterations: 12

# Final equation : log(p) = 8.094 - 8.616\*V1 - 3.948\*V2 - 5.217\*V3  
  
# Confusion matrices   
p1 <- predict(m1.best, train, type = 'response')  
pred1 <- ifelse(p1>0.5,1,0)  
cm1 <- table(pred1, train$V5) # for training data  
cm1

##   
## pred1 0 1  
## 0 377 4  
## 1 4 301

p1 <- predict(m1.best, test, type = 'response')  
pred1 <- ifelse(p1>0.5,1,0)  
cm2 <- table(pred1, test$V5) # for testing data  
cm2

##   
## pred1 0 1  
## 0 377 3  
## 1 4 302

# Misclassification Error/Accuracy  
errorTrain <- (377+301)/(377+301+8) # 98.8% / 1.2%  
errorTest <- (377+302)/(377+301+7) # 99.1% / 1.9%  
  
# Specificity, sensitivity and accuracy  
Spec1 <- cm1[1]/(cm1[1]+cm1[3]) # 0.989  
sens1 <- cm1[4]/(cm1[2]+cm1[4]) #0.986  
   
Spec2 <- cm2[1]/(cm2[1]+cm2[3]) # 0.992  
sens2 <- cm2[4]/(cm2[2]+cm2[4]) # 0.986

**The equation for calculating if the specimen is genuine is given by the best model:**

**Here log(p) is the log of probability that the sample is genuine.**

**With a misclassification error of 1.2% and 1.9% for training and testing data respectively, we can say that the model performs very well.**

**It also has very high specificity and sensitivity for both the testing and training data.**

1. (40 points) Develop logistic regression models with 60%/40%, 70%/30%, and 80%/20% partitioning into training and testing data sets using set.seed(222). Summarize training and testing accuracy, sensitivity and specificity for each in the table below and compare with 50%/50% performance using the table below. Recommend and comment on the best model for future use based on model accuracy. (No need to reproduce codes here as they are similar to part 1 and 2.)

|  |  |  |  |
| --- | --- | --- | --- |
| **Partitioning** | **Accuracy %** | **Sensitivity %** | **Specificity %** |
| Training - 50% | 98.8 | 98.6 | 98.9 |
| Testing – 50% | 99.1 | 98.6 | 99.2 |
| Training - 60% | 99.6 | 99.4 | 99.7 |
| Testing – 40% | 98.7 | 98.7 | 98 |
| Training - 70% | 99.5 | 99.5 | 99.6 |
| Testing – 30% | 97.8 | 98.4 | 97.1 |
| Training - 80% | 99.1 | 99.1 | 99.1 |
| Testing – 20% | 98.9 | 99.1 | 98.7 |

1. (20 points) Compare the best and the worst logistic regression model in the previous question using ROC curve, AUC and best threshold values based on testing data. Discuss your results.

r <- multiclass.roc(test$V5, p1, percent = TRUE)

## Setting direction: controls < cases

roc <- r[['rocs']]  
r1 <- roc[[1]]  
plot.roc(r1,print.auc = T,  
 print.thres = T,  
 auc.polygon = T,  
 grid = c(0.1,0.2),  
 main = "ROC curve for 50:50 split")

Chart

Description automatically generated

r <- multiclass.roc(test$V5, p3, percent = TRUE)

## Setting direction: controls < cases

roc <- r[['rocs']]  
r3 <- roc[[1]]  
plot.roc(r3,print.auc = T,  
 print.thres = T,  
 auc.polygon = T,  
 grid = c(0.1,0.2),  
 main = "ROC curve for 70:30 split")

Chart

Description automatically generated

**Based on testing data, we can conclude that the split 50:50 (model 1) performs best in terms of accuracy, while the worst performing model is the one with 70:30 (model 2) split. Even though the difference is not large, with model 1 having AUC=100% and model 2 having AUC=99.9%**

**Both are good ROC curves because they hug the top left corner of the graph.**

**The best threshold for model 1 is 0.6 whereas for model 2 it is 0.8.**

**DELIVERABLE**

1. (15 points) Create R Markdown file for this assignment and knit it in a Word format. Submit a single PDF based on the knitted file covering all five questions, code, output and comments.