Question_No_8

Utsab Bhattarai

2024-05-31

Do the following in R studio using "Arrests" dataset of car package with R script to knit PDF output:

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3
```

```
data("Arrests")
```

a. Divide the Arrests data into train and test datasets with 80:20 random splits.

```
set.seed(35)
sample <- sample(c(TRUE, FALSE),
                 replace = TRUE,
                 prob = c(0.7, 0.3))
trainData <- Arrests[sample, ]
testData <- Arrests[!sample, ]
```

b. Fit a supervised logistic regression and Naive Bayes classification models on train data with "released" as dependent variable and colour, age, sex, employed and citizen as independent variable.

```
# Fit a supervised logistic regression
logit_model <- glm(released ~ colour + age + sex + employed + citizen,
                   data = trainData,
                   family = binomial)
summary(logit_model)
```

```
##
## Call:
## glm(formula = released ~ colour + age + sex + employed + citizen,
##     family = binomial, data = trainData)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.404415   0.295651   1.368 0.171349
## colourWhite  0.562533   0.116248   4.839  1.3e-06 ***
## age         -0.009615   0.006115  -1.572 0.115873
## sexMale     -0.196211   0.197465  -0.994 0.320394
## employedYes  1.016575   0.114528   8.876  < 2e-16 ***
## citizenYes   0.505232   0.137499   3.674 0.000238 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2428.8  on 2612  degrees of freedom
## Residual deviance: 2279.5  on 2607  degrees of freedom
## AIC: 2291.5
##
## Number of Fisher Scoring iterations: 4
```

```r
# Fit the Naive Bayes model
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```r
nb_model <- naiveBayes(released ~ colour + age + sex + employed + citizen,
                       data = trainData)
nb_model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        No       Yes
## 0.1756602 0.8243398
##
## Conditional probabilities:
##      colour
## Y         Black     White
##   No  0.3812636 0.6187364
##   Yes 0.2200557 0.7799443
##
##      age
## Y         [,1]     [,2]
##   No  25.02614 9.092644
```

```
##   Yes 23.53900 8.191825
##
##      sex
## Y         Female       Male
##   No  0.07625272 0.92374728
##   Yes 0.09470752 0.90529248
##
##      employed
## Y            No       Yes
##   No  0.3899782 0.6100218
##   Yes 0.1727019 0.8272981
##
##      citizen
## Y            No       Yes
##   No  0.2352941 0.7647059
##   Yes 0.1244197 0.8755803
```

c. Predict the released variable in the test datasets of these models and interpret the result carefully.

```
# Predict on test data using logistic regression
# logit_preds <- predict(logit_model,
                        # newdata = testData,
                        # type = "response")
# logit_class <- ifelse(logit_preds > 0.5,
                      # 1,
                      # 0)
# Confusion matrix for logistic regression
# library(caret)
# logit_cm <- confusionMatrix(factor(logit_class),
                             # factor(testData$released))
# logit_cm
# Predict on test data using Naive Bayes
# nb_preds <- predict(nb_model,
                    # newdata = testData)
# Confusion matrix for Naive Bayes
# nb_cm <- confusionMatrix(nb_preds,
                          # factor(testData$released))
# nb_cm
```

d. Compare and decide which classification model is better for this data.

```
# Comparing models based on accuracy and other metrics
# logit_accuracy <- logit_cm$overall['Accuracy']
# nb_accuracy <- nb_cm$overall['Accuracy']

# Comparing other metrics if necessary
# logit_cm
# nb_cm
```