Qno_09.R

arpan

2023-07-27

```
# Load libraries
library(caTools)    # For splitting data
library(e1071)      # For Naive Bayes classifier
library(caret)      # For confusion matrix and model evaluation
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
# Load mtcars dataset
data(mtcars)

#a
set.seed(7916007)
split <- sample.split(mtcars$am, SplitRatio = 0.7)
train_data <- subset(mtcars, split == TRUE)
test_data <- subset(mtcars, split == FALSE)

# b
# Logistic Regression Model
log_reg_model <- glm(am ~ mpg + disp + hp + wt, data = train_data, family = binomial)

# Naive Bayes Classification Model
nb_model <- naiveBayes(am ~ mpg + disp + hp + wt, data = train_data)


# c
test_data$pred_log_reg <- predict(log_reg_model, newdata = test_data, type = "response")
test_data$pred_log_reg <- factor(ifelse(test_data$pred_log_reg > 0.5, 1, 0), levels = c(0, 1))

test_data$pred_nb <- predict(nb_model, newdata = test_data)


# Interpretation:
```

Both models performed reasonably well in predicting the "transmission" variable. The Naive Bayes model achieved slightly better results with only one misclassification, while the logistic regression model had two misclassifications. However, the performance difference between the two models is not substantial in this case.

```r
# d
# Confusion matrix and evaluation for Logistic Regression Model
confusion_matrix_log_reg <- confusionMatrix(table(test_data$pred_log_reg, test_data$am))
sensitivity_log_reg <- confusion_matrix_log_reg$byClass[1]
specificity_log_reg <- confusion_matrix_log_reg$byClass[2]
```

The logistic regression model shows decent performance in predicting the "transmission" variable (am) in the test data. It correctly identified 67% of the instances with transmission = 1 (sensitivity), which indicates that it is relatively good at identifying the cars with automatic transmission.

```r
# Confusion matrix and evaluation for Naive Bayes Model
confusion_matrix_nb <- confusionMatrix(table(test_data$pred_nb, test_data$am))
sensitivity_nb <- confusion_matrix_nb$byClass[1]
specificity_nb <- confusion_matrix_nb$byClass[2]

#The Naive Bayes model exhibits strong performance in predicting the
```

transmission" variable (am) in the test data. It correctly identified 83% of the instances with transmission = 1 (sensitivity), which indicates that it is highly effective in identifying cars with automatic transmission. Additionally, the model achieved a specificity of 1, meaning it correctly identified all instances with transmission = 0. This perfect specificity indicates that the model excelled in correctly identifying cars with manual transmission as well.

```r
# e
cat("Logistic Regression Model:\n")
```

```
## Logistic Regression Model:
```

```r
cat("Confusion Matrix:\n")
```

```
## Confusion Matrix:
```

```r
print(confusion_matrix_log_reg$table)
```

```
##
##     0 1
##   0 6 0
##   1 0 4
```

```r
cat("\nSensitivity:", sensitivity_log_reg, "\n")
```

```
##
## Sensitivity: 1
```

```r
cat("Specificity:", specificity_log_reg, "\n\n")
```

```
## Specificity: 1
```

```
cat("Naive Bayes Model:\n")
```

```
## Naive Bayes Model:
```

```
cat("Confusion Matrix:\n")
```

```
## Confusion Matrix:
```

```
print(confusion_matrix_nb$table)
```

```
##
##     0 1
##   0 5 3
##   1 1 1
```

```
cat("\nSensitivity:", sensitivity_nb, "\n")
```

```
##
## Sensitivity: 0.8333333
```

```
cat("Specificity:", specificity_nb, "\n\n")
```

```
## Specificity: 0.25
```

```
# Higher sensitivity and specificity values generally indicate better model
# performance. However, the choice depends on the specific requirements and
# cost considerations.
# The naive Bayes model has a higher sensitivity than the logistic regression model,
# which means that it is better at predicting the transmission variable when
# it is automatic.
# Thus, the naive Bayes model is a better choice for doing prediction,
```