Build and Evaluate Logistic Regression Model

Nithya Sarabudla

03-03-2024

Question 1

```
# Load the dataset "student-mat.csv"
student_mat <- read.csv("/Users/nithyasarabudla/DA5030/student-mat.csv",sep = ";", header = TRUE, string
```

Question 2

```
# Adding a new column 'PF' to the data frame 'student_mat'
# This column marks students with a final grade ('G3') less than 10 as 'F' (Fail) and others as 'P' (Pa
student_mat$PF <- ifelse(student_mat$G3 < 10, "F", "P")
# Creating a new column 'PF_dummy' in the 'student_mat' data frame
# This column is a dummy variable where 'F' (Fail) is encoded as 0 and 'P' (Pass) is encoded as 1
student_mat$PF_dummy <- ifelse(student_mat$PF == "F", 0, 1)
```

Question 3

```
# Building an initial logistic regression model to predict pass/fail status
# 'PF_dummy' is the response variable, and 'age', 'absences', 'G1', 'G2', 'studytime' are predictors
# The 'binomial' family with a 'logit' link function is specified because this is a binary classificati
initial_logistic_model <- glm(PF_dummy ~ absences + age + G1 + G2 + studytime, data = student_mat, famil

# Optimizing the initial model by backward elimination of non-significant predictors
# The 'step' function iteratively removes the least significant variable at each step, based on AIC
final_logistic_model <- step(initial_logistic_model, direction = "backward")
```

```
## Start:  AIC=146.09
## PF_dummy ~ absences + age + G1 + G2 + studytime
##
##             Df Deviance    AIC
## <none>           134.09 146.09
## - absences   1   136.45 146.45
## - G1         1   138.96 148.96
## - age        1   139.39 149.39
## - studytime  1   141.22 151.22
## - G2         1   236.08 246.08
```

```r
# Displaying the summary of the final logistic model, including coefficients, significance levels, and
summary(final_logistic_model)
```

```
##
## Call:
## glm(formula = PF_dummy ~ absences + age + G1 + G2 + studytime,
##     family = binomial(link = "logit"), data = student_mat)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.62299    3.66254  -2.900  0.00373 **
## absences     -0.03672    0.02688  -1.366  0.17187
## age          -0.40373    0.18222  -2.216  0.02672 *
## G1            0.33769    0.15526   2.175  0.02963 *
## G2            1.76739    0.28265   6.253 4.03e-10 ***
## studytime    -0.76204    0.29769  -2.560  0.01047 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.50  on 394  degrees of freedom
## Residual deviance: 134.09  on 389  degrees of freedom
## AIC: 146.09
##
## Number of Fisher Scoring iterations: 8
```

Question 4

```r
# Create the regression equation by pasting together the coefficients from the final logistic regressio
Regression_equation <- paste("log(p / (1 - p)) = ", paste(final_logistic_model$coefficients, collapse =
# Print the regression equation
cat("Regression Equation: ", Regression_equation, "\n")
```

```
## Regression Equation:  log(p / (1 - p)) =  -10.6229877587987 + -0.036720037637179 + -0.40372777624410
```

Question 5

```r
# Generate predictions using the final logistic regression model on the entire dataset
pred <- ifelse(predict(final_logistic_model, type = "response") > 0.5, 1, 0)
# Calculate the accuracy of the model by comparing predictions with actual values
accu <- mean(pred == student_mat$PF_dummy)
# Print the model accuracy
cat("Model Accuracy: ", accu, "\n")
```

```
## Model Accuracy:  0.9189873
```

The accuracy of the logistic regression model is 0.9189873

Question 6

```r
# Load the necessary library for kNN
library(class)

# Select the features for kNN prediction
knn <- student_mat[, c("absences", "age", "G1", "G2")]

# Set seed for reproducibility
set.seed(42)

# Split the data into training and testing sets (70% training, 30% testing)
training_indices <- sample(1:nrow(knn), 0.7 * nrow(knn))
training_data <- knn[training_indices, ]
testing_data <- knn[-training_indices, ]

# Train the kNN model using the training data and predict the testing data
knn_model <- knn(training_data, testing_data, cl = student_mat$PF_dummy[training_indices], k = 5)

# Calculate the accuracy of the kNN model by comparing predictions with actual values
knn_accu <- mean(knn_model == student_mat$PF_dummy[-training_indices])

# Print the accuracy of the kNN model
cat("kNN Model Accuracy: ", knn_accu, "\n")
```

```
## kNN Model Accuracy:  0.8739496
```

The accuracy of kNN model is 0.8739496

Question 7

The logistic regression model achieved an accuracy of 0.9189873, while the kNN model achieved an accuracy of 0.8739496.User The logistic regression model accuracy is slightly higher when compared to the kNN model on this dataset.

The difference between the two algorithms lies in their approach to classification:

Logistic Regression is a linear model for binary classification that predicts the probability of an instance belonging to a default class. It works well with linearly separable data and can handle relationships between features. It's relatively simple and provides interpretable results, making it a good choice for problems where the relationship between features is approximately linear.

k-Nearest Neighbors (kNN) is a non-parametric, instance-based learning algorithm where the classification of an instance is determined by the majority vote of its neighbors, with the instance being assigned to the class most common among its k nearest neighbors. kNN works well with a small number of input variables , but struggles with high-dimensional data.

When deciding between logistic regression and kNN, consider the following: logistic regression is preferred for linear decision boundaries and offers interpretability through feature coefficients, making it suitable for larger, high-dimensional datasets. In contrast, kNN excels in scenarios with non-linear decision boundaries but can be computationally intensive and less effective in high-dimensional spaces. Logistic regression provides quick predictions once trained, whereas kNN's performance depends on the dataset's size and dimensionality.