```r
#----------------------------Initial Setup--------------------------

# Set random number seed for reproducibility
set.seed(7406)

# Set the working directory
setwd("C:/Users/ns14555/Desktop/Projects/002 Project 2 (HW3)/003 Project
work")

# Check the current working directory to confirm it was set correctly
getwd()

# Load required libraries
library(ggplot2)
library(MASS)
library(glmnet)
library(lars)
library(pls)
library(caret)
library(e1071)
library(nnet)
library(class)
library(gridExtra)
library(tidyr)
#-----------------------Data Loading and Preparation-------------------

# Load Data
auto_data <- read.table("C:/Users/ns14555/Desktop/Projects/002 Project 2
(HW3)/003 Project work/Auto.csv",head=T,
                        sep=",")

#Median value of mpg column
mpg_median <- median(auto_data$mpg)
mpg_median
#22.75
#mpg01 column creation
auto_data$mpg01 <- ifelse(auto_data$mpg > mpg_median, 1, 0)
#Remove mpg column
auto_data <- subset(auto_data, select = -mpg)
#Getting mpg01 column as the first column
auto_data <- auto_data[,c("mpg01",setdiff(names(auto_data), "mpg01"))]
```

```r
# Exploratory Data Analysis
# Dimension
dim(auto_data)
# 392 rows 8 columns
head(auto_data)
#counting the number oif mp01 values of 1 and 0
sum(auto_data[,1]==1)
#196
sum(auto_data[,1]==0)
#196
summary(auto_data)

#Defining response and predictor variables
response_variable <- 'mpg01'
predictor_variables <- c("cylinders", "displacement", "horsepower",
                         "weight", "acceleration", "year","origin" )

# List of numerical columns to plot
numerical_cols <- c("displacement", "horsepower", "weight", "acceleration",
"year")

# Histograms for each numerical column
histogram_list <- lapply(numerical_cols, function(col) {
  ggplot(auto_data, aes_string(x = col)) +
    geom_histogram(fill = "skyblue", color = "black", bins = 30) +
    theme_minimal() +
    theme(panel.grid.major = element_line(color = "gray", size = 0.5),
          panel.grid.minor = element_line(color = "gray", size = 0.25)) +
    labs(title = paste("Histogram of", col), x = col, y = "Frequency")
})

# Combine plots into a grid
grid.arrange(grobs = histogram_list, ncol = 2)

# Scatterplots
scatterplots <- lapply(predictor_variables, function(pred_var) {
  ggplot(auto_data, aes_string(x = pred_var, y = response_variable)) +
    geom_point() +
    labs(title = paste("Scatterplot of", pred_var, "vs",
                       response_variable))
})
# Convert scatterplots to grob objects
scatterplots_grobs <- lapply(scatterplots, ggplotGrob)
```

```
# Arrange scatterplots in a grid
grid.arrange(grobs = scatterplots_grobs, ncol = 2)

# Boxplots
# Reshape the data from wide to long format using tidyr
auto_data_long <- pivot_longer(auto_data, cols = c(cylinders, displacement,
                                                   horsepower, weight,
                                                   acceleration, year,
origin))
# Create the boxplot with customized appearance
ggplot(auto_data_long, aes(x = name, y = value)) +
  geom_boxplot(fill = "gray", color = "black", outlier.color = "red",
               fatten = 2) + # 'fatten' increases the width of the median
line
  facet_wrap(~ name, scales = "free", ncol = 4) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),  #for better
visibility
        axis.title.x = element_blank(), # Remove x-axis title
        axis.title.y = element_blank()) # Remove y-axis title

#Redefining response and predictor variables
response_variable <- 'mpg01'
predictor_variables <- c("displacement", "horsepower", "weight",
                         "acceleration" )
## Another look at the first several rows
dim(auto_data)
#392 5

n = dim(auto_data)[1] # total number of observations
n1 = round(n/10) # number of observations randomly selected for testing
data
# Split the data into training and test set
train <- sample(c(TRUE, FALSE), nrow(auto_data),replace=TRUE, prob=c(0.8,
0.3))
auto_train <- auto_data[train, ]
auto_test <- auto_data[!train,]

dim(auto_train) ## Distribution of the data labels in the training data
#285   8
dim(auto_test) ## Distribution of the data labels in the testing data
#107   8

#------------------------Model Building----------------------
```

```r
# head(auto_train)
# head(auto_train[,2:5])
# head(auto_train[,1])
# head(y_true)
# Initialize testing error collection
train_MSE <- NULL
test_MSE <- NULL

# Model 1:Linear Discriminant Analysis (LDA)
model_lda <- lda(auto_train[,-1], auto_train[,1])
pred_lda_train <- predict(model_lda, auto_train[,-1])$class
train_MSE <- round(c(train_MSE, mean(pred_lda_train!=auto_train$mpg01)),8)
pred_lda_test <- predict(model_lda, auto_test[,-1])$class
test_MSE <- round(c(test_MSE, mean((pred_lda_test!=auto_test$mpg01))),8)

# Model 2:Quadratic Discriminant Analysis (QDA)
model_qda <- qda(auto_train[,-1], auto_train[,1])
pred_qda_train <- predict(model_qda, auto_train[,-1])$class
train_MSE <- round(c(train_MSE, mean(pred_qda_train!=auto_train$mpg01)),8)
pred_qda_test <- predict(model_qda, auto_test[,-1])$class
test_MSE <- round(c(test_MSE, mean((pred_qda_test!=auto_test$mpg01))),8)

# Model 3: Naive Bayes
model_naiveBayes <- naiveBayes(auto_train[,-1], auto_train[,1])
pred_naiveBayes_train <- predict(model_naiveBayes, auto_train[,-1])
train_MSE <- round(c(train_MSE,
mean(pred_naiveBayes_train!=auto_train$mpg01)),8)
pred_naiveBayes_test <- predict(model_naiveBayes, auto_test[,-1])
test_MSE <- round(c(test_MSE,
mean((pred_naiveBayes_test!=auto_test$mpg01))),8)

# Model 4: Multinomial logisitic regression
model_lr <- multinom(mpg01~., , data=auto_train)
pred_lr_train <- predict(model_lr, auto_train[,-1])
train_MSE <- round(c(train_MSE, mean(pred_lr_train!=auto_train$mpg01)),8)
pred_lr_test <- predict(model_lr, auto_test[,-1])
test_MSE <- round(c(test_MSE, mean((pred_lr_test!=auto_test$mpg01))),8)

# Model 5: KNN with several values
k_list <- c(1,2,3,4,5,6,7,8,9,10);
xnew <- auto_train[,-1];
xnew2 <- auto_test[,-1];
train_errors <-NULL
```

```
test_errors <-NULL
for (i in 1: 8){
  kk <- k_list[i];
  pred4 <- knn(auto_train[,-1], xnew, auto_train[,1], k=kk);
  train_errors <- rbind( train_errors, cbind(kk,
                                             mean( pred4 !=
auto_train[,1])));
  pred4.test <- knn(auto_train[,-1], xnew2, auto_train[,1], k=kk);
  test_errors <- rbind( test_errors, cbind(kk,
                                           mean( pred4.test!=
                                                   auto_test[,1])));
}
results <- data.frame(
  K = train_errors[, 1],  # K-values
  Train_Error = train_errors[, 2],  # Training errors
  Test_Error = test_errors[, 2]  # Testing errors
)
results
#k=5 is the best k value
model_knn <- knn(auto_train[,-1], auto_train[,-1], auto_train[,1], k=5)
train_MSE <- round(c(train_MSE, mean(model_knn!=auto_train$mpg01)),8)
pred_knn_test <- knn(auto_train[,-1], auto_test[,-1], auto_train[,1], k=5)
test_MSE <- round(c(test_MSE, mean((pred_knn_test!=auto_test$mpg01))),8)

#Tables for models and Errors
# Table for k values with training and testing errors
models <-c('LDA','QDA','Naive Bayes','Logistic Regression','KNN')
model_results <- data.frame(
  Models = models,
  Train_Error = train_MSE,
  Test_Error = test_MSE
)
model_results

#--------------------Monte Carlo Cross-Validation--------------------

set.seed(7407) # Reset seed fr reproducibility
B <- 100
TEALL <- matrix(nrow = B, ncol =5) # Preallocate matrix for efficiency

for (b in 1:B){
  #-------------Initial Preparation------------
  indices <- sample(1:nrow(auto_data), size = round(0.2*nrow(auto_data)))
```

```r
  train_data <- auto_data[-indices,]
  test_data <- auto_data[indices,]

  #------------- Model Building-------------------
  # Model 1:Linear Discriminant Analysis (LDA)
  model_1 <- lda(train_data[,-1], train_data[,1])
  pred_1 <- predict(model_1, test_data[,-1])$class
  te1 <- round(mean((pred_1!=test_data$mpg01)),8)

  # Model 2:Quadratic Discriminant Analysis (QDA)
  model_2 <- qda(train_data[,-1], train_data[,1])
  pred_2 <- predict(model_2, test_data[,-1])$class
  te2 <- round(mean((pred_2!=test_data$mpg01)),8)

  # Model 3: Naive Bayes
  model_3 <- naiveBayes(train_data[,-1], train_data[,1])
  pred_3 <- predict(model_3, test_data[,-1])
  te3<- round(mean((pred_3!=test_data$mpg01)),8)

  # Model 4: Multinomial logisitic regression
  model_4 <- multinom(mpg01~., , data=train_data)
  pred_4 <- predict(model_4, test_data[,-1])
  te4 <- round(mean((pred_lr_test!=auto_test$mpg01)),8)

  # Model 5: KNN with several values
  k_list <- c(1,2,3,4,5,6,7,8,9,10);
  xnew <- train_data[,-1];
  xnew2 <- test_data[,-1];
  train_errors <-NULL
  test_errors <-NULL
  for (i in 1: 8){
    kk <- k_list[i];
    pred5 <- knn(train_data[,-1], xnew, train_data[,1], k=kk);
    train_errors <- rbind( train_errors, cbind(kk,
                                             mean( pred5 !=
train_data[,1])));
    pred5.test <- knn(train_data[,-1], xnew2, train_data[,1], k=kk);
    test_errors <- rbind( test_errors, cbind(kk,
                                           mean( pred5.test!=
                                                   test_data[,1])));
  }
  results_k <- data.frame(
    K = train_errors[, 1],  # K-values
```

```
    Train_Error = train_errors[, 2],  # Training errors
    Test_Error = test_errors[, 2]  # Testing errors
  )
  # results
  #k=3 is the best k value
  pred_5 <- knn(train_data[,-1], test_data[,-1], train_data[,1], k=3)
  te5 <- round(mean((pred_5!=test_data$mpg01)),8)


  #Collect the testing errors
  TEALL = cbind(te1, te2, te3, te4, te5)
}
results_k
#best k value = 3
TEALL

colnames(TEALL) <- c('LDA','QDA','Naive Bayes','Logistic Regression','KNN')
#Before Cross Validation
model_results
#After Cross Validation
TEALL

# From the mean standard error, we found the LDA to be the best

#------------------------------ The End -------------------------------
```