MTH443 Quiz 2

Email: nkaushik20@iitk.ac.in

Kaushik Raj Nadar (208160499)

2024-11-07

Problem Statement

Consider the dataset quiz2.csv having the following variables:

Variable	Description
age	Age of the patient in years
cp	Chest pain type: 0: Typical angina, 1: Atypical angina, 2: Non-anginal pain, 3: Asymptomatic
trestbps	Resting blood pressure in mm Hg
chol	Serum cholesterol in mg/dl
fbs	Fasting blood sugar level, categorized as above 120 mg/dl (1 = true, 0 = false)
restecg	Resting electrocardiographic results: 0: Normal, 1: Having ST-T wave abnormality, 2: Showing probable or definite left ventricular hypertrophy
thalach	Maximum heart rate achieved during a stress test
exang	Exercise-induced angina (1 = yes, 0 = no)
oldpeak	ST depression induced by exercise relative to rest
ca	Number of major vessels (0-4) colored by fluoroscopy
HDS	Heart disease status (0 = no disease, 1 = presence of disease)

Load necessary libraries

```
library(tidyverse)
library(ggplot2)
library(MASS)
```

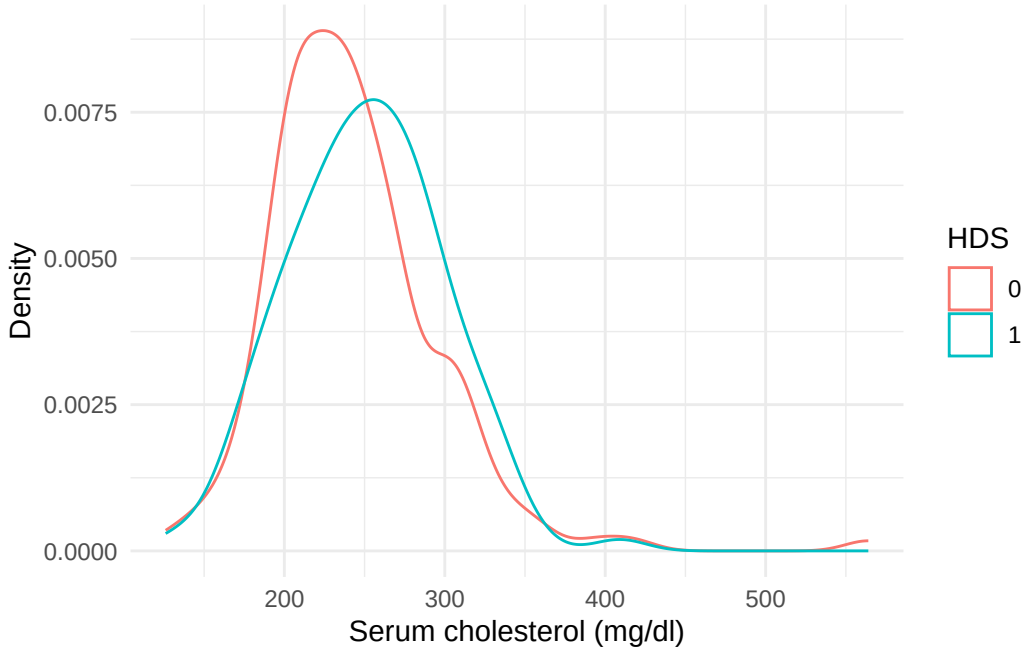
Load the dataset

```r
dataset <- read.csv("quiz2.csv")

# Clean the dataset
dataset <- dataset %>% filter(dataset$HDS == 1 | dataset$HDS==0)

# Print Head of Dataset
head(dataset)
```

```
  age cp trestbps chol fbs restecg thalach exang oldpeak ca HDS
1  63  0      145  233   1       2     150     0     2.3  0   0
2  67  3      160  286   0       2     108     1     1.5  3   1
3  67  3      120  229   0       2     129     1     2.6  2   1
4  37  2      130  250   0       0     187     0     3.5  0   0
5  41  1      130  204   0       2     172     0     1.4  0   0
6  56  1      120  236   0       0     178     0     0.8  0   0
```

(a) Obtain density estimate plots of the variable "chol" for HDS value 0 group and for HDS value 1 group using kernel density estimation method with a Gaussian kernel.

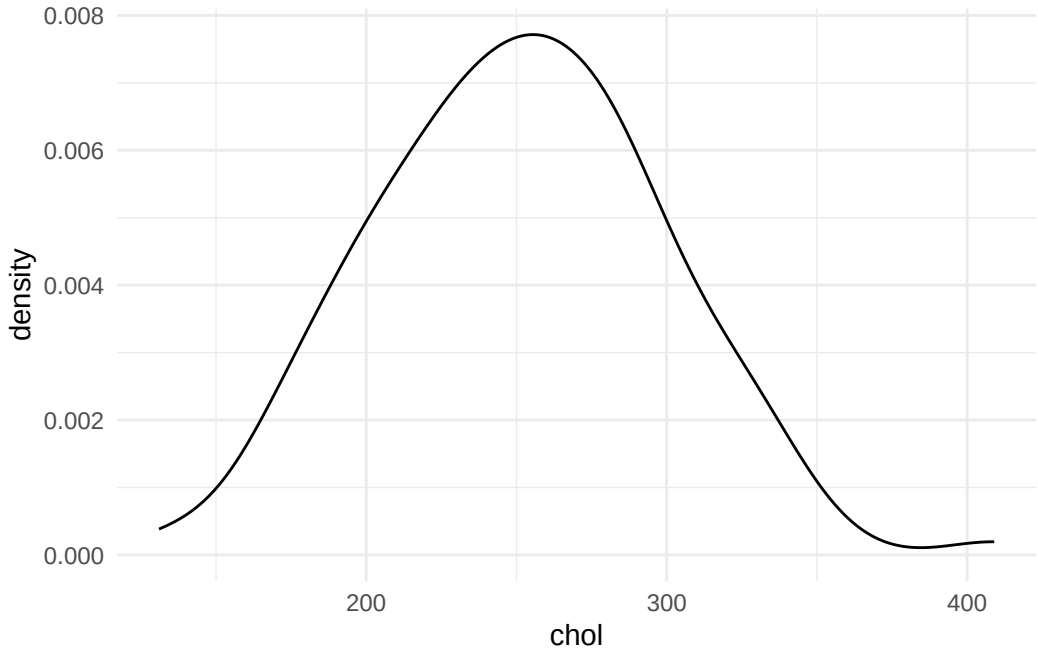
```r
dataset %>%
  ggplot(aes(x = chol, color = factor(HDS))) +
  geom_density(kernel = "gaussian") +
  labs(x = "Serum cholesterol (mg/dl)", y = "Density", color = "HDS") +
  theme_minimal()
```



(b) Using the estimated density obtained in (a), find P(chol > 250) for the group with HDS value 1.

Plot Density Estimate for HDS Value=1

```
hds1_density <- dataset %>%
  filter(HDS == 1) %>%
  ggplot(aes(x = chol)) +
  geom_density(kernel = "gaussian") +
  theme_minimal()

print(hds1_density)
```



Calculate the P(chol > 250)

```
chol_HDS_1 <- dataset$chol[dataset$HDS == 1]

# Kernel density estimation for chol in HDS = 1 group
density_HDS_1 <- density(chol_HDS_1, kernel = "gaussian")

# Approximate probability P(chol > 250) using the density estimate
P_chol_gt_250 <- sum((density_HDS_1$y[density_HDS_1$x > 250])
                * diff(density_HDS_1$x)[1])
print(paste("P(chol > 250) for HDS = 1:", round(P_chol_gt_250, 4)))
```

```
[1] "P(chol > 250) for HDS = 1: 0.5096"
```

A probability of 0.51 suggests that slightly more than half of the individuals with heart disease in this dataset have cholesterol levels exceeding 250 mg/dl.

This result indicates a notable association between elevated cholesterol levels and the presence of heart disease, as a cholesterol level above 250 mg/dl is relatively common among those with heart disease in this data sample.

(c) Split the available data into 2 parts, keeping the last 10% records as out of sample test data and apply the following classifiers to build classification models for the 2-class problem with classification variable as HDS: (i) linear discriminant function, (ii) quadratic discriminant function.

```
# Split the data into training and test sets
set.seed(123)
n <- nrow(dataset)
test_size <- 0.1
test_idx <- sample(1:n, size = round(n * test_size))
train_data <- dataset[-test_idx, ]
test_data <- dataset[test_idx, ]

# (i) Linear Discriminant Function
lda_model <- lda(HDS ~ ., data = train_data)
lda_pred_train <- predict(lda_model, train_data)$class
lda_pred_test <- predict(lda_model, test_data)$class

# (ii) Quadratic Discriminant Function
qda_model <- qda(HDS ~ ., data = train_data)
qda_pred_train <- predict(qda_model, train_data)$class
qda_pred_test <- predict(qda_model, test_data)$class
```

(d) Calculate the misclassification error rates of the classification models obtained in (c), separately for the learning data and the test set data.

```
# Training and test error for LDA
lda_train_error <- mean(lda_pred_train != train_data$HDS)
lda_test_error <- mean(lda_pred_test != test_data$HDS)
print(paste("LDA Training Error Rate:", round(lda_train_error, 4)))
```

```
[1] "LDA Training Error Rate: 0.1907"
```

```
print(paste("LDA Test Error Rate:", round(lda_test_error, 4)))
```

```
[1] "LDA Test Error Rate: 0.2069"
```

```
# Training and test error for QDA
qda_train_error <- mean(qda_pred_train != train_data$HDS)
qda_test_error <- mean(qda_pred_test != test_data$HDS)
print(paste("QDA Training Error Rate:", round(qda_train_error, 4)))
```

```
[1] "QDA Training Error Rate: 0.1829"
```

```
print(paste("QDA Test Error Rate:", round(qda_test_error, 4)))
```

```
[1] "QDA Test Error Rate: 0.1724"
```

QDA appears to perform better than LDA on both the training and test datasets, as indicated by its lower error rates. This suggests that **a quadratic decision boundary is more appropriate** for this data than a linear one.

Both models show a slight increase in error from the training set to the test set, which is expected as models usually perform better on data they were trained on. However, the difference in error rates is not large, suggesting neither model is severely overfitting.

Given the lower test error rate of the QDA model, it might be preferable to use QDA for future predictions in this context, as it seems to capture the underlying structure of the data more accurately than LDA.