HeartDisease_Assignment

Venkata Sai Reddy

2025-03-11

Install required packages

```
# install.packages("caret")
# install.packages("e1071")
# install.packages("dplyr")
# install.packages("tinytex")
```

Loading required libraries

```
library(caret)  # For machine learning and confusion matrix
```

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

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

```
library(e1071)  # For Naive Bayes classifier
library(dplyr)  # For data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tinytex)  # For PDF rendering
```

Q1: Predicting Heart Disease Risk for Patients with Chest Pain

```
## Load the dataset

Heart_disease <- read.csv("C:/Users/sunny/Downloads/Heart_disease.csv")

dim(Heart_disease) # Checking the dimensions
```

```
## [1] 303   9
```

```
colnames(Heart_disease)# Displaying the column names
```

```
## [1] "Age"                "Sex"                "chest_pain_type"
## [4] "Blood_Pressure"     "Cholestrol"         "Fasting_Blood_Sugar"
## [7] "Rest_ECG"           "MAX_HeartRate"      "Exercise"
```

```
head(Heart_disease)
```

```
##   Age Sex chest_pain_type Blood_Pressure Cholestrol Fasting_Blood_Sugar
## 1  63   1               0            145        233                   1
## 2  37   1               1            130        250                   0
## 3  41   0               1            130        204                   0
## 4  56   1               1            120        236                   0
## 5  57   0               0            120        354                   0
## 6  57   1               0            140        192                   0
##   Rest_ECG MAX_HeartRate Exercise
## 1        0           150        0
## 2        1           187        0
## 3        0           172        0
## 4        1           178        0
## 5        1           163        1
## 6        1           148        0
```

```
summary(Heart_disease) # summary of the dataset
```

```
##       Age             Sex         chest_pain_type  Blood_Pressure
##  Min.   :29.00   Min.   :0.0000   Min.   :0.0000   Min.   : 94.0
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :0.0000   Median :130.0
##  Mean   :54.37   Mean   :0.6832   Mean   :0.4521   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :1.0000   Max.   :200.0
##    Cholestrol    Fasting_Blood_Sugar    Rest_ECG      MAX_HeartRate
##  Min.   :126.0   Min.   :0.0000      Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000      1st Qu.:0.0000   1st Qu.:133.5
##  Median :240.0   Median :0.0000      Median :1.0000   Median :153.0
##  Mean   :246.3   Mean   :0.1485      Mean   :0.5281   Mean   :149.6
##  3rd Qu.:274.5   3rd Qu.:0.0000      3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000      Max.   :2.0000   Max.   :202.0
##     Exercise
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3267
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Interpretation

1. The dataset has 9 columns (attributes) and 303 rows (patients), each of which represents a distinct heart health-related characteristic.

2.The columns include information such as age, sex, blood pressure, cholesterol, chest pain type, exercise capacity, Rest_ECG, Fasting_Blood_Sugar and MAX_HeartRate.

```
# Create the 'Target' dummy variable

Heart_disease <- Heart_disease %>%
  mutate(Target = ifelse(MAX_HeartRate > 170, "Yes", "No"))

dim(Heart_disease)  # Checking the dimensions

## [1] 303  10

colnames(Heart_disease)# Displaying the column names

##  [1] "Age"                "Sex"                "chest_pain_type"
##  [4] "Blood_Pressure"     "Cholestrol"         "Fasting_Blood_Sugar"
##  [7] "Rest_ECG"           "MAX_HeartRate"      "Exercise"
## [10] "Target"
```

Interpretation

To classify patients with a maximal heart rate higher than 170 as “Yes” (signaling a higher cardiac risk or abnormality) and “No” otherwise, a new “Target” variable was developed.

“Target” is a new column added in the dataset

```
# Create the 'BP_New' dummy variable
Heart_disease <- Heart_disease %>%
  mutate(BP_New = ifelse(Blood_Pressure > 120, "Yes", "No"))

dim(Heart_disease) # Checking the dimensions

## [1] 303  11

colnames(Heart_disease) # Displaying the column names

##  [1] "Age"                "Sex"                "chest_pain_type"
##  [4] "Blood_Pressure"     "Cholestrol"         "Fasting_Blood_Sugar"
##  [7] "Rest_ECG"           "MAX_HeartRate"      "Exercise"
## [10] "Target"             "BP_New"
```

Interpretation

A new variable called “BP_New” was added, classifying patients as “Yes” (elevated blood pressure) if their blood pressure was higher than 120 and “No” otherwise.

This new column has been added to the dataset and can be used for analysis or to monitor patient blood pressure levels.

```
# Creating  a table for the 'Target' variable
Target_table <- table(Heart_disease$Target)
```

```r
# Displaying the table
print(Target_table)
```

```
##
##  No Yes
## 245  58
```

Interpretation

245 patients have a “No” value for the “Target” variable (meaning their maximum heart rate is not greater than 170).

58 patients have a “Yes” value for the “Target” variable (indicating their maximum heart rate is greater than 170).

```r
# Determine the most frequent outcome
most_frequent_target <- names(which.max(Target_table))

# Print the prediction based on initial information
cat("Prediction for heart disease based on initial information:",
most_frequent_target)
```

```
## Prediction for heart disease based on initial information: No
```

Interpretation

“No” is the most frequent response for the “Target” variable, indicating that most patients do not have a maximal heart rate higher than 170 based on the initial data.

Q2: Analysis of the First 30 Records

a) Compute Bayes Conditional Probabilities

```r
# Select the first 30 records from the dataset
Heart_disease30 <- Heart_disease[1:30, c("Target", "BP_New",
"chest_pain_type")]

# Checking the dimensions of the selected subset Heart_disease30
dim(Heart_disease30)
```

```
## [1] 30  3
```

```r
# Displaying the column names of the selected subset  Heart_disease30 to
confirm the variables
colnames(Heart_disease30)
```

```
## [1] "Target"          "BP_New"          "chest_pain_type"
```

Interpretation

Using three variables—whether the individual has heart disease (Target), their blood pressure status (BP_New), and the type of chest discomfort (chest_pain_type)—I have chosen the first 30 records from the dataset.

```
# Create pivot tables (contingency tables)

Object1 <- ftable(Heart_disease30)  # Pivot table for all variables
Object2 <- ftable(Heart_disease30[, -1])  # Pivot table excluding 'Target'

# Check the structure of the pivot tables
dim(Object1)

## [1] 4 2

dim(Object2)

## [1] 2 2

print(Object1)

##               chest_pain_type 0 1
## Target BP_New
## No     No                     2 2
##        Yes                    7 8
## Yes    No                     0 3
##        Yes                    3 5

print(Object2)

##        chest_pain_type  0  1
## BP_New
## No                      2  5
## Yes                    10 13
```

Interpretation

The association between the target variable (Target), blood pressure status (BP_New), and kind of chest discomfort (chest_pain_type) is displayed in the pivot tables Object1 and Object2.

Target is not included in Object2, yet all three variables are in Object1.

The values show how various combinations of these variables (e.g., no heart disease with chest pain type = 1 and BP_New = Yes) have eight records.

The conditional probabilities for every combination are computed with the aid of the contingency tables.

```r
# Correct indexing for the conditional probabilities
P1 <- Object1[3, 1] / Object2[1, 1]  # BP_New = No, chest_pain_type = 0 and
Target = Yes
P2 <- Object1[3, 2] / Object2[1, 2]  # BP_New = No, chest_pain_type = 1 and
Target = Yes
P3 <- Object1[4, 1] / Object2[2, 1]  # BP_New = Yes, chest_pain_type = 0 and
Target = Yes
P4 <- Object1[4, 2] / Object2[2, 2]  # BP_New = Yes, chest_pain_type = 1 and
Target = Yes

# Display the conditional probabilities
prob_results <- rbind(P1, P2, P3, P4)
colnames(prob_results) <- c("Conditional Probability")
rownames(prob_results) <- c("P1", "P2", "P3", "P4")

# Print the probabilities
print(prob_results)
```

```
##    Conditional Probability
## P1               0.0000000
## P2               0.6000000
## P3               0.3000000
## P4               0.3846154
```

Interpretation

P1 (BP_New = No, chest_pain_type = 0): When blood pressure is No and chest pain type is 0, the conditional probability of having heart disease (Target = Yes) is 0.000. This suggests that no heart disease patients were found in this combination.

P2 (BP_New = No, chest_pain_type = 1): When blood pressure is zero and chest pain type is one, the conditional probability of having heart disease is 0.600. This indicates that this group has a comparatively greater risk of heart disease.

P3 (BP_New = Yes, chest_pain_type = 0): When blood pressure is Yes and chest pain type is 0, there is a 0.300 conditional probability of heart disease. This suggests that there is a moderate chance of heart disease in this situation.

P4 (BP_New = Yes, chest_pain_type = 1): When blood pressure is Yes and chest pain type is 1, the conditional chance of getting heart disease is 0.3846. This indicates that this group has a modest risk of heart disease.

These odds aid in our comprehension of how the chance of heart disease is influenced by the combination of blood pressure and the type of chest discomfort.

b) Classification of the 30 Records

```r
# Initialize a vector to store the predicted probabilities
Probability_Target <- rep(0, 30)
```

```r
# Assign probabilities based on the conditions for each record
for (i in 1:30) {
  BP_New_value <- Heart_disease30$BP_New[i]
  chest_pain_value <- Heart_disease30$chest_pain_type[i]

# Determine which conditional probability applies for each combination of
predictors

  if (BP_New_value == "No" & chest_pain_value == 0) {
    Probability_Target[i] <- P1
  } else if (BP_New_value == "No" & chest_pain_value == 1) {
    Probability_Target[i] <- P2
  } else if (BP_New_value == "Yes" & chest_pain_value == 0) {
    Probability_Target[i] <- P3
  } else if (BP_New_value == "Yes" & chest_pain_value == 1) {
    Probability_Target[i] <- P4
  }
}

# Add the predicted probabilities to the dataset
Heart_disease30$Probability_Target <- Probability_Target

# Classify each record based on the 0.5 threshold for the target variable
Heart_disease30$Pred_Probability <- ifelse(Heart_disease30$Probability_Target
> 0.5, "Yes", "No")

# Display the results of the classification
print(Heart_disease30)
```

```
##    Target BP_New chest_pain_type Probability_Target Pred_Probability
## 1      No    Yes               0          0.3000000               No
## 2     Yes    Yes               1          0.3846154               No
## 3     Yes    Yes               1          0.3846154               No
## 4     Yes     No               1          0.6000000              Yes
## 5      No     No               0          0.0000000               No
## 6      No    Yes               0          0.3000000               No
## 7      No    Yes               1          0.3846154               No
## 8     Yes     No               1          0.6000000              Yes
## 9      No    Yes               1          0.3846154               No
## 10    Yes    Yes               1          0.3846154               No
## 11     No    Yes               0          0.3000000               No
## 12     No    Yes               1          0.3846154               No
## 13    Yes    Yes               1          0.3846154               No
## 14     No     No               0          0.0000000               No
## 15     No    Yes               0          0.3000000               No
## 16     No     No               1          0.6000000              Yes
## 17    Yes     No               1          0.6000000              Yes
## 18     No    Yes               0          0.3000000               No
## 19    Yes    Yes               0          0.3000000               No
```

```
## 20     No    Yes                0          0.3000000                 No
## 21     No    Yes                0          0.3000000                 No
## 22    Yes    Yes                1          0.3846154                 No
## 23    Yes    Yes                0          0.3000000                 No
## 24     No    Yes                1          0.3846154                 No
## 25    Yes    Yes                0          0.3000000                 No
## 26     No    Yes                1          0.3846154                 No
## 27     No    Yes                1          0.3846154                 No
## 28     No     No                1          0.6000000                Yes
## 29     No    Yes                1          0.3846154                 No
## 30     No    Yes                1          0.3846154                 No
```

Interpretation

The model uses the parameters for BP_New and chest_pain_type to predict the probability of heart disease (0.3 to 0.6) for each record.

If the estimated likelihood of heart disease is more than 0.5, the data is classified as “Yes”; otherwise, it is classified as “No.”

Few records in the given data have a higher likelihood (0.6), which leads to a “Yes.” The majority of the records have a low expected probability (about 0.3), which leads to a “No” classification.

c) Manual Calculation of Naive Bayes Probability

```
# Compute the prior probability for Target = Yes

prob_target_yes <- sum(Heart_disease30$Target == "Yes") /
nrow(Heart_disease30)
print(prob_target_yes)
```

```
## [1] 0.3666667
```

```
# Compute the probability of BP_New = Yes and chest_pain_type = 1, given
Target = Yes

Total_BP_Chest_Yes <- sum(Heart_disease30$Target == "Yes" &
Heart_disease30$BP_New == "Yes" & Heart_disease30$chest_pain_type == 1)
P_BP_Chest_Yes <- Total_BP_Chest_Yes / sum(Heart_disease30$Target == "Yes")
print(P_BP_Chest_Yes)
```

```
## [1] 0.4545455
```

```
# Compute the probability of BP_New = Yes and chest_pain_type = 1 (marginal
probability)

P_BP_Chest_1 <- sum(Heart_disease30$BP_New == "Yes" &
Heart_disease30$chest_pain_type == 1) / nrow(Heart_disease30)
print(P_BP_Chest_1)
```

```
## [1] 0.4333333

# Calculate Naive Bayes probability for Target = Yes, given BP_New = Yes and
chest_pain_type = 1

Bayes_Prob <- (P_BP_Chest_Yes * prob_target_yes) / P_BP_Chest_1

# Display the Naive Bayes probability

cat("Naive Bayes Probability of Target = Yes given BP_New = Yes and
chest_pain_type = 1: ", Bayes_Prob, "\n")

## Naive Bayes Probability of Target = Yes given BP_New = Yes and
chest_pain_type = 1:  0.3846154
```

Interpretation

Prior likelihood (Target = Yes): About 0.367, or 36.7% of the records, had a prior likelihood of having heart disease (Target = Yes).

Conditional Probability: Given that the patient has heart disease, the likelihood of having both high blood pressure (BP_New = Yes) and type 1 chest discomfort is 0.455.

Marginal Probability: There is a 0.433 chance of experiencing both type 1 chest discomfort and high blood pressure.

Naive Bayes likelihood: Based on these numbers, a person with high blood pressure and type 1 chest discomfort has a 38.5% Naive Bayes likelihood of having heart disease.

Q3: Naive Bayes Classification and Performance Evaluation on Heart Disease Dataset

```
# Set seed for reproducibility
set.seed(123)

# Splitting the data into training and validation sets

train_indices <- sample(row.names(Heart_disease), 0.6 *
dim(Heart_disease)[1])
validation_indices <- setdiff(row.names(Heart_disease), train_indices)
train_dataset <- Heart_disease[train_indices, ]
validation_dataset <- Heart_disease[validation_indices, ]


# Ensure class distribution is maintained in both sets

cat("Class distribution in Training Set:\n")
```

```
## Class distribution in Training Set:

print(table(train_dataset$Target))

##
##  No Yes
## 149  32

cat("Class distribution in Validation Set:\n")

## Class distribution in Validation Set:

print(table(validation_dataset$Target))

##
##  No Yes
##  96  26

# Check for duplicates between training and validation sets

duplicates_in_both <- intersect(row.names(train_dataset),
row.names(validation_dataset))



# Print results

if (length(duplicates_in_both) == 0) {
  cat("No duplicates found between training and validation datasets.\n")
} else {
  cat("Warning: Duplicates found in both sets!\n")
  print(duplicates_in_both)
}

## No duplicates found between training and validation datasets.
```

Interpretation

There are 149 “No” and 32 “Yes” examples in the training set and 96 “No” and 26 “Yes” cases in the validation set. This makes the model training more dependable by guaranteeing that both sets maintain the initial class proportions.

Additionally, we look for duplicates, which indicates that there are no data points that overlap between the training and validation sets, indicating that the dataset split was completed correctly and that no data was lost.

```
# Train Naive Bayes model using relevant categorical predictors
nb_model <- naiveBayes(Target ~ chest_pain_type + BP_New, data =
train_dataset)
```

```r
# Print the model summary
print(nb_model)
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        No       Yes
## 0.8232044 0.1767956
##
## Conditional probabilities:
##      chest_pain_type
## Y          [,1]      [,2]
##   No  0.3892617 0.4892273
##   Yes 0.7187500 0.4568034
##
##      BP_New
## Y            No       Yes
##   No  0.3288591 0.6711409
##   Yes 0.3750000 0.6250000
```

Interpretation

According to the model, there is a class imbalance because 82.34% of the cases fit into the “No” group (no disease), but only 17.6% fall into the “Yes” category (disease present).

71.9% of people with the condition (“Yes”) fall into a certain category of chest discomfort, indicating that it is a significant predictor.

Additionally, 32.9% of those with heart disease (“Yes”) had normal blood pressure, whereas 67.11 have high blood pressure. 37.5% of people without heart disease (“No”) have normal blood pressure, while 62.5% have high blood pressure.

```r
# Make predictions on the training data
train_predictions <- predict(nb_model, train_dataset)

# Convert to factors for correct evaluation
train_dataset$Target <- factor(train_dataset$Target)
train_predictions <- factor(train_predictions, levels =
levels(train_dataset$Target))


print(train_predictions)
```

```
##   [1] No No No No No No No No No No No No No No No No No No No No No No No
No No
```

```
##  [26] No No No No No No No No No No No No No No No No No No No No No No No
No No
##  [51] No No No No No No No No No No No No No No No No No No No No No No No
No No
##  [76] No No No No No No No No No No No No No No No No No No No No No No No
No No
## [101] No No No No No No No No No No No No No No No No No No No No No No No
No No
## [126] No No No No No No No No No No No No No No No No No No No No No No No
No No
## [151] No No No No No No No No No No No No No No No No No No No No No No No
No No
## [176] No No No No No No
## Levels: No Yes
```

Interpretation

With the majority of people being categorized as “No” (having no heart disease), the predicted values closely resemble the actual values, indicating that the model performs well on the training data.

```
# Creating and displaying the confusion matrix
# Compute the confusion matrix
conf_matrix_train <- confusionMatrix(train_predictions, train_dataset$Target)
print(conf_matrix_train)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  149  32
##        Yes   0   0
##
##                Accuracy : 0.8232
##                  95% CI : (0.7597, 0.8758)
##     No Information Rate : 0.8232
##     P-Value [Acc > NIR] : 0.5471
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 4.251e-08
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.8232
##          Neg Pred Value :    NaN
##              Prevalence : 0.8232
##          Detection Rate : 0.8232
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
```

```
##
##        'Positive' Class : No
##
```

Interpretation

With an accuracy rate of 82.32%, the model properly classifies most situations.

“No” Class Perfect Sensitivity It is quite dependable for this category because it can properly identify all “No” cases (100% sensitivity).