FML Assignment 4

Pooja Babu

2025-03-03

Loading all the libraries

```
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(readr)

library(e1071)

library(pROC)

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(naivebayes)

## Warning: package 'naivebayes' was built under R version 4.4.3

## naivebayes 1.0.0 loaded

## For more information please visit:

## https://majkamichal.github.io/naivebayes/
```

```
library(dplyr)

##
## 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)

## Warning: package 'tinytex' was built under R version 4.4.3
```

Importing the dataset

```
heart <- read_csv("C:/Users/pooja/OneDrive/Desktop/FML Assignments/4th
Assignment/Heart_disease.csv")

## Rows: 303 Columns: 9
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (9): Age, Sex, chest_pain_type, Blood_Pressure, Cholestrol,
Fasting_Bloo...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

Checking the dimensions of the dataset

```
dim(heart) #The dataset consists of 303 observations with 9 variables.

## [1] 303   9
```

Checking the structure of the dataset

```
str(heart)

## spc_tbl_ [303 × 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Age              : num [1:303] 63 37 41 56 57 57 56 44 52 57 ...
##  $ Sex              : num [1:303] 1 1 0 1 0 1 0 1 1 1 ...
##  $ chest_pain_type  : num [1:303] 0 1 1 1 0 0 1 1 1 1 ...
##  $ Blood_Pressure   : num [1:303] 145 130 130 120 120 140 140 120 172
150 ...
##  $ Cholestrol       : num [1:303] 233 250 204 236 354 192 294 263 199
168 ...
##  $ Fasting_Blood_Sugar: num [1:303] 1 0 0 0 0 0 0 0 1 0 ...
##  $ Rest_ECG         : num [1:303] 0 1 0 1 1 1 0 1 1 1 ...
##  $ MAX_HeartRate    : num [1:303] 150 187 172 178 163 148 153 173 162
```

```
174 ...
##  $ Exercise            : num [1:303] 0 0 0 0 1 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Age = col_double(),
##   ..   Sex = col_double(),
##   ..   chest_pain_type = col_double(),
##   ..   Blood_Pressure = col_double(),
##   ..   Cholestrol = col_double(),
##   ..   Fasting_Blood_Sugar = col_double(),
##   ..   Rest_ECG = col_double(),
##   ..   MAX_HeartRate = col_double(),
##   ..   Exercise = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Checking for missing values in the dataset

```
sum(is.na(heart))

## [1] 0
```

Checking for the duplicates in the dataset

```
sum(duplicated(heart))

## [1] 1
```

Removing the duplicates in the dataset

```
heart_new<-heart[!duplicated(heart),]
```

Creating the required dummy variables

```
heart_new$Target<-ifelse(heart_new$MAX_HeartRate > 170, "Yes","No")

heart_new$BP_New<- ifelse(heart_new$Blood_Pressure>120,"Yes","No")
```

Question 1: Prediction Based on Initial Information

```
 Target_table <- table(heart_new$Target)

print(Target_table)

## 
##  No Yes 
## 245  57

# Calculating the probability of "Yes" i.e having a heart disease if the
heart rate is more than 170.
probability_yes <- Target_table["Yes"]/sum(Target_table)
probability_yes

##       Yes 
## 0.1887417

# Calculating the probability of "No" i.e having a heart disease if the heart
rate is less than or equal to 170.
probability_no<-Target_table["No"]/sum(Target_table)
probability_no

##        No 
## 0.8112583

# Interpretation for target table output
# A new dummy variable "Target" was created where individuals who had their
heart rate more than 170 were marked "Yes" and the individuals who had their
heart rate below or equal to 170 were marked "No". Based on this dataset, we
have to predict if a person presents only with Chest pain and no additional
information, whether they are likely to have a heart disease or not.Based on
the outputs, we can see that out of 302 individuals 245 individuals are those
whose heart rate is less than 170 and the probability of not having a heart
disease is 81%. And the remaining individuals i.e 57 of them are those whose
heart rate is more than 170 and the probability of them having a heart
disease is 19%.

# So going with the highest probability, considering only the chest pain as a
predictor, we can see that most of the Individuals do not have a heart
disease.
```

Question 2: Analysis of the first 30 records

```
# Selecting the first 30 records in all the three datasets.
heart_new30<-
heart_new%>%slice(1:30)%>%select("Target","BP_New","chest_pain_type")

# Creating a pivot table with all the three variables "Target", "Chest
Pain","BP New"
```

```
object1 <- ftable(heart_new30)
object1

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

# Creating a pivot table without target column
object2<-ftable(heart_new30$BP_New,heart_new30$chest_pain_type)
object2

##       0  1
##
## No    2  5
## Yes  10 13

# A. Compute Bayes Conditional Probabilities:

#Number of cases with heart disease ("No") where BP_New = No and
chest_pain_type = 0.
# Total number of cases where BP_New = No and chest_pain_type = 0.

p1 = object1[3,1]/object2[1,1]

# Probability of heart disease given BP is normal and chest pain type = 0.
p1

## [1] 0

# Heart disease cases ("Yes") where BP_New = Yes and chest_pain_type = 0.
# Total cases with BP_New = No and chest_pain_type = 1.

p2 = object1[4,1]/object2[1,2]

# Probability of heart disease given BP is normal and chest pain type = 1.
p2

## [1] 0.6

# Heart disease cases ("No") where BP_New = No and chest_pain_type = 1.
# Total cases with BP_New = Yes and chest_pain_type = 0.

p3 = object1[3,2]/object2[2,1]

# Probability of heart disease given high BP and chest pain type is 0.
p3

## [1] 0.3
```

```
# Heart disease cases ("Yes") where BP_New = Yes and chest_pain_type = 1.
# Total cases with BP_New = Yes and chest_pain_type = 1.

p4 = object1[4,2]/object2[2,2]

# Probability of heart disease given high BP and chest pain type = 1.
p4
```

```
## [1] 0.3846154
```

```
# B. Classification of Accidents:

prob_target<-rep(0,30)
for(i in 1:30){          # Creating a for loop to iterate through each record
from 1 to 30.

bp_value<-heart_new30$BP_New[i]

chest_value<-heart_new30$chest_pain_type[i]

if(bp_value =="Yes" & chest_value == 0 ){prob_target[i]<-p1}#Assigning p1,If
BP is Yes and chest pain type is 0.

else if(bp_value =="No" & chest_value == 1 ){prob_target[i]<-p2}#Assigning
p2,If BP is No and chest pain type is 1.

else if(bp_value == "Yes" & chest_value == 1){prob_target[i]<-p3}#Assigning
p3,If BP is Yes and chest pain type is 1.

else(prob_target[i]<-p4)#Assigning p4 for all the other cases
}

# Assigns the previously calculated probability of heart disease
(prob_target) to a new column prob_target in heart_new30.

heart_new30$prob_target<-prob_target

# Creating a new column pred_probability in heart_new30 using ifelse() to
classify each individual as either "Yes"or"No" (heart disease predicted or
not) if prob_target > 0.5, classify as "Yes" (high probability of heart
disease) Otherwise, classify as "No" (low probability of heart disease).

heart_new30$pred_probability <- ifelse(heart_new30$prob_target >
0.5,"Yes","No")
```

```
heart_new30

## # A tibble: 30 × 5
##    Target BP_New chest_pain_type prob_target pred_probability
##    <chr>  <chr>            <dbl>       <dbl> <chr>
##  1 No     Yes                  0       0     No
##  2 Yes    Yes                  1       0.3   No
##  3 Yes    Yes                  1       0.3   No
##  4 Yes    No                   1       0.6   Yes
##  5 No     No                   0       0.385 No
##  6 No     Yes                  0       0     No
##  7 No     Yes                  1       0.3   No
##  8 Yes    No                   1       0.6   Yes
##  9 No     Yes                  1       0.3   No
## 10 Yes    Yes                  1       0.3   No
## # ℹ 20 more rows



# C. Manual Calculation of Naive Bayes Probability:

# Calculating the total number of rows in the datasets.
tt_count <-nrow(heart_new30)

# Calculating how many individuals have heart disease i.e Target = Yes
divided by total count.
prob_target_yes <-sum(heart_new30$Target == "Yes")/tt_count

# Calculating cases where BP_New is "Yes" , chest_pain_type = 1 and target =
"Yes" divided by total number of individuals with Target = "Yes"
given_yes_1_yes <-sum(heart_new30$BP_New == "Yes" &
heart_new30$chest_pain_type == 1 & heart_new30$Target ==
"Yes")/sum(heart_new30$Target == "Yes")

# Calculating how many individuals have BP_New = "Yes" and chest_pain_type =
1 divided by the total count
given_yes_1 <-sum(heart_new30$BP_New == "Yes" & heart_new30$chest_pain_type
== 1)/tt_count

# Calculating the Final Probability Using 'Bayes' Theorem
probability_yes_given_bpchest<- (given_yes_1_yes * prob_target_yes) /
given_yes_1

# Printing the calculated probability of heart disease given high blood
pressure and chest pain.

cat("Calculating the naive Bayes conditional probability of an injury
manually given that BP_New is Yes and chest_pain_type is 1 = ",
probability_yes_given_bpchest, "\n")

## Calculating the naive Bayes conditional probability of an injury manually
given that BP_New is Yes and chest_pain_type is 1 =  0.3846154
```

Question 3. Full Dataset Analysis

```
#Splitting the data into 60% training and 40% validation.

set.seed(123) # Setting seed for reproducibility

index_train<-sample(row.names(heart_new),0.6*dim(heart_new)[1])
index_valid<-setdiff(row.names(heart_new),index_train)
training <- heart_new[index_train,]
validation <- heart_new[index_valid,]

# Checking the number of rows for training and validation dataset.
nrow(training)

## [1] 181

nrow(validation)

## [1] 121

# The "Exercise" variable does not significantly contribute to predicting
heart disease,so removing it to simplify the model.
training<-training[,-9]
validation<-validation[,-9]

# Ensuring the target variable in the datasets to be factors.
validation$Target<-as.factor(validation$Target)
training$Target<-as.factor(training$Target)

# training a Naive Bayes classifier on the training dataset using the
naiveBayes() function where target is the response variable and
chest_pain_type and BP_New are the predictor variables. Setting laplace = 1
so that no probability is completely zero
naivebayes_model1 <- naiveBayes(Target ~ chest_pain_type + BP_New, data =
training , laplace = 1)

# predicting the validation dataset using the trained model.
validation_pred<-predict(naivebayes_model1,validation)

# Ensuring validation_pred has the same factor levels as the actual Target
variable.
validation$Target<-factor(validation$Target)
validation_pred<-factor(validation_pred,levels=levels(validation$Target))
```

```r
# Creating a confusion matrix for predicted values vs actual values
confusionMatrix(validation_pred,validation$Target,positive = "Yes")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction No Yes
##        No  96  25
##        Yes  0   0
##                                           
##                Accuracy : 0.7934          
##                  95% CI : (0.7103, 0.8616)
##     No Information Rate : 0.7934          
##     P-Value [Acc > NIR] : 0.5533          
##                                           
##                   Kappa : 0               
##                                           
##  Mcnemar's Test P-Value : 1.587e-06       
##                                           
##             Sensitivity : 0.0000          
##             Specificity : 1.0000          
##          Pos Pred Value :    NaN          
##          Neg Pred Value : 0.7934          
##              Prevalence : 0.2066          
##          Detection Rate : 0.0000          
##    Detection Prevalence : 0.0000          
##       Balanced Accuracy : 0.5000          
##                                           
##        'Positive' Class : Yes             
## 
```

```r
# Interpretation for naivebayes_model1 :
# The naivebayes model was trained initially using training dataset to
predict the "Target" variable with only BP_New and Chest_pain_type as the
predictor variables. When the model was tested on the validation dataset, the
accuracy was 79% and the AUC was 56% which means the model is performing very
slightly better than the random model.The first model is not a reliable one
as per the outcome.




# training a Naive Bayes classifier on the training dataset using the
naiveBayes() function where target is the response variable and all the other
columns in the dataset are the predictor variables. Setting laplace = 1 so
that no probability is completely zero.
naivebayes_model2 <- naiveBayes(Target ~., data = training , laplace = 1)
```

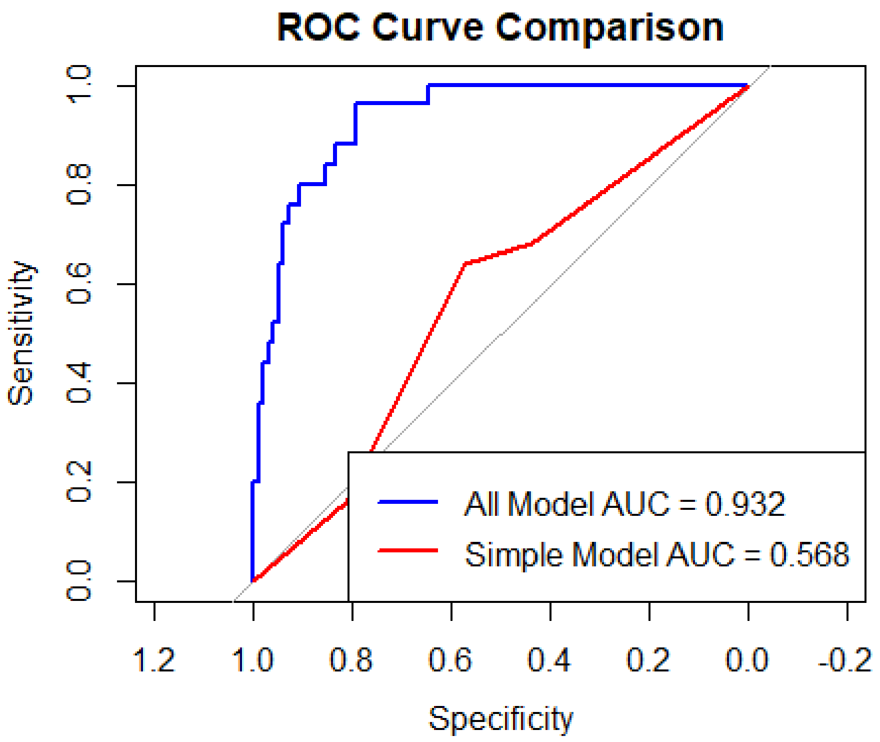
```
# predicting the validation dataset using the trained model.
pred1<-predict(naivebayes_model2,validation)
pred1<-factor(pred1,levels=levels(validation$Target))

# Creating a confusion matrix for predicted values vs actual values
confusionMatrix(pred1,validation$Target,positive = "Yes")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction No Yes
##        No  90   8
##        Yes  6  17
##                                           
##                Accuracy : 0.8843          
##                  95% CI : (0.8135, 0.9353)
##     No Information Rate : 0.7934          
##     P-Value [Acc > NIR] : 0.006446        
##                                           
##                   Kappa : 0.6363          
##                                           
##  Mcnemar's Test P-Value : 0.789268        
##                                           
##             Sensitivity : 0.6800          
##             Specificity : 0.9375          
##          Pos Pred Value : 0.7391          
##          Neg Pred Value : 0.9184          
##              Prevalence : 0.2066          
##          Detection Rate : 0.1405          
##    Detection Prevalence : 0.1901          
##       Balanced Accuracy : 0.8088          
##                                           
##        'Positive' Class : Yes             
## 
```

```
# Interpretation for naivebayes_model2:
# The naivebayes  model 2 was trained initially using training dataset to
predict the "Target" variable with all the remaining columns as predictor
variables. When the model was tested on validation dataset, the accuracy was
88% and the AUC value was 93% which means the model is performing very well
for unseen data. And the accuracy was increased when we include other
variables also while predicting the heart disease. So these factors also play
an important role in predicting the heart disease in real world cases.
```

```
# Predicted probabilities for all variables as predictors.
pred_probs_all <- predict(naivebayes_model2, validation, type = "raw")
prob_yes_all <- pred_probs_all[, "Yes"]
```

```
# Predicted probabilities for Reduced Model where only BP_New and
Chest_pain_type were predictor variables.
pred_probs_simple <- predict(naivebayes_model1, validation, type = "raw")
prob_yes_simple <- pred_probs_simple[, "Yes"]

# Computing the ROC curves
roc_all <- roc(validation$Target, prob_yes_all)

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

roc_simple <- roc(validation$Target, prob_yes_simple)

## Setting levels: control = No, case = Yes
## Setting direction: controls < cases

# Plot Full Model ROC curve
plot(roc_all, col = "blue", lwd = 2, main = "ROC Curve Comparison")
legend("bottomright", legend = c("All Model", "Simple Model"), col =
c("blue", "red"), lwd = 2)

# Adding the simplified Model ROC curve
lines(roc_simple, col = "red", lwd = 2)

# Adding AUC values to legend of the ROC graph
auc_all <- auc(roc_all)
auc_simple <- auc(roc_simple)
legend("bottomright", legend = c(paste("All Model AUC =", round(auc_all, 3)),
                                 paste("Simple Model AUC =",
round(auc_simple, 3))),
       col = c("blue", "red"), lwd = 2)
```



```
# This is just a simple graph comparing the ROC Curves for naivebayes_model1
and naivebayes_model2. The AUC value for naivebayes_model1 is 0.568 which
means the model cannot be relied on whereas the AUC value for
naivebayes_model2 is 0.932 which means the model is performing very good
compared to the random model.
```