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))

# 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   
##

# 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)

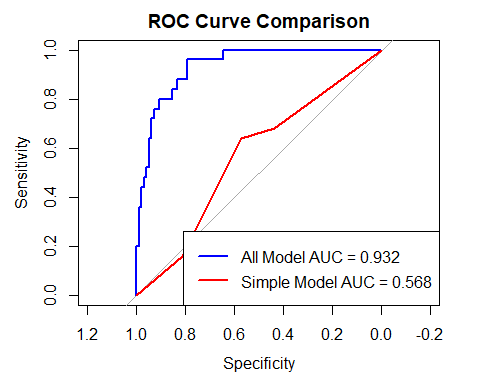
## Setting levels: control = No, case = Yes

## 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.