ML_Assignment_4

Jacob Joy

Setup

```
#call Packages
library(naivebayes)
```

```
## Warning: package 'naivebayes' was built under R version 4.4.1

## naivebayes 1.0.0 loaded

## For more information please visit:

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

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.1

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.4.1

## Loading required package: lattice
```

```
library(pivottabler)
```

```
## Warning: package 'pivottabler' was built under R version 4.4.1
```

```
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.4.1
```

```
#Import Data
heart_Data = read.csv("C:\\Heart_disease.csv")

#Set Seed for consistency
set.seed(123)
```

```r
#Creating Dummy variable for Target Yes if MAX HeartRate
#greater than 170 else NO
for(i in 1:length(heart_Data)){
  heart_Data$Target[i] = ifelse(heart_Data$MAX_HeartRate[i] > 170, "YES", "NO")
}

#Creating Dummy variable for BP_New if Blood_Pressure is greater
#than 120 else NO
for(i in 1:length(heart_Data)){
  heart_Data$BP_New[i] = ifelse(heart_Data$Blood_Pressure[i] > 120, "YES", "NO")
}
```

Setup Information

I preformed the above setup, which included created dummy variables for two numerical variables for use in the Naive Bayes Classifier because continuous numeric variables do not work well with this model.

Question 1

```r
#Created a table to compare the target for heart disease with if they had
#chest pain
Target_table = table(heart_Data$Target, heart_Data$chest_pain_type,
                     dnn = c("Heart Disease", "Chest Pain"))

#Created the table to display probabilities instead of just frequency count
prop.table(Target_table)
```

```
##              Chest Pain
## Heart Disease          0          1
##           NO  0.54785479 0.43894389
##           YES 0.00000000 0.01320132
```

Question 1 Explanation

Based on only the information on the table created if a person presented with Chest Pain the most likely outcome would be that they do not have Heart Disease. However, because chest pain can be life threatening a doctor would still want to verify on more then the symptom of Chest Pain.

Question 2 A Code

```r
#Create a new dataframe with the first 30 entries which only includes the
#features Target, BP_New, and chest_pain_type
heart_data30 = heart_Data[1:30, c("Target", "BP_New", "chest_pain_type")]

#Creation of piviot table using all three variables
```

```
pivot1 = ftable(heart_data30)
pivot1
```

```
##                chest_pain_type  0  1
## Target BP_New
## NO     NO                       1  0
##        YES                     11 14
## YES    NO                       0  2
##        YES                      0  2
```

```
#Creation of each probability
P_YES_YES_1 = nrow(subset(heart_data30, Target == "YES" & BP_New == "YES"
                          & chest_pain_type == 1)) / nrow(heart_data30)

P_YES_NO_0 = nrow(subset(heart_data30, Target == "YES" & BP_New == "NO" &
                           chest_pain_type == 0)) / nrow(heart_data30)

P_YES_NO_1 = nrow(subset(heart_data30, Target == "YES" & BP_New == "NO" &
                           chest_pain_type == 1)) / nrow(heart_data30)

P_YES_YES_0 = nrow(subset(heart_data30, Target == "YES" & BP_New == "YES" &
                            chest_pain_type == 0)) / nrow(heart_data30)

P_YES_1 = nrow(subset(heart_data30, BP_New == "YES" &
                        chest_pain_type == 1)) / nrow(heart_data30)

P_NO_0 = nrow(subset(heart_data30, BP_New == "NO" &
                       chest_pain_type == 0)) / nrow(heart_data30)

P_NO_1 = nrow(subset(heart_data30, BP_New == "NO" &
                       chest_pain_type == 1)) / nrow(heart_data30)

P_YES_0 = nrow(subset(heart_data30, BP_New == "YES" &
                        chest_pain_type == 0)) / nrow(heart_data30)

#Creation of the variables that have the conditional probabilities
#with the outcome being yes for target.

P1 = P_YES_YES_0 / P_YES_0
P2 = P_YES_YES_1 / P_YES_1
P3 = P_YES_NO_0 / P_NO_0
P4 = P_YES_NO_1 / P_NO_1

print(paste("The conditional probability of having heart disease with high BP and No chest Pain", P1))
```

```
## [1] "The conditional probability of having heart disease with high BP and No chest Pain 0"
```

```
print(paste("The conditional probability of having heart disease with high BP and chest Pain", P2))
```

```
## [1] "The conditional probability of having heart disease with high BP and chest Pain 0.125"
```

```
print(paste("The conditional probability of having heart disease withregular BP and No chest Pain", P3)
```

```
## [1] "The conditional probability of having heart disease withregular BP and No chest Pain 0"
```

```
print(paste("The conditional probability of having heart disease with high BP and chest Pain", P4))
```

```
## [1] "The conditional probability of having heart disease with high BP and chest Pain 1"
```

Question 2 A Explination

In question 2 A I created a new dataframe of only the first 30 observations and then created a pivot table based on this information. Next, I created the probabilities for each outcome of variables but only if the Target variable was Yes. Using the created probabilities I used the formula for conditional probability to find the 4 different variations that could occur. Based on just these probabilities the “only” indication of heart disease is having chest pain.

Question 2 B Code

```
#Created a numeric variable of 30 entries all set to 0
Probability_Target = rep(0,30)

#Using a for loop and an if statement each observation in heart_data30 was
#checked and given a probability based on the conditional probabilities
#calculated in Part A, because the only variable that resulted in heart_disease
#was the presence of chest pain any other option would be set to 0

for (i in 1:30){
  if (heart_data30$BP_New[i] == "YES" & heart_data30$chest_pain_type[i] == 1){
    Probability_Target[i] = P2
    } else if (heart_data30$BP_New[i] == "NO" &
               heart_data30$chest_pain_type[i] == 1){

      Probability_Target[i] = P4
    } else {
      Probability_Target[i] = 0
    }
}
#Adding the Probability_Target variable to our dataframe
heart_data30$Probability_Target = Probability_Target

#Using the Probability_Target and a cutoff value of 0.5 a prediction was
#made if the person would have heart disease or not

heart_data30$Pred_Probability =
  ifelse(heart_data30$Probability_Target > 0.5, "Yes", "No")

table(heart_data30$Target, heart_data30$Pred_Probability)
```

```
##
##        No Yes
##   NO   26   0
##   YES   2   2
```

Question 2 B Explination

After completing Step B and creating a table to see the results from our target compared with the Predicted classification we can see that with a cutoff value of 0.5 there would have been a potential to miss 2 people that had heart disease. Again because of the nature of this data and peoples lives the cutoff value would want to be lowered drastically to make sure that people with chest pain are evaluated. (This is true in the real world, as your doctor will send you to the ER if you have chest pain)

Question 2 C

```
#Created the probability of having heart disease
P_YES = nrow(subset(heart_data30, Target == "YES")) / nrow(heart_data30)

#Using the case of a observation having high BP and Chest Pain,
#I used Bayes theorem to find the probability
naive = (P_YES_YES_1 * P_YES) / P_YES_1
naive
```

```
## [1] 0.01666667
```

Question 2 C Explination

The probability found using Bayes Theorem of a person having heart disease who has high BP and Chest pain is 1.67% This makes sense based on the data we have and problems conducted before this. The chances of heart disease is low over all and when looking at only 2 specific events we see that this is even lower. This also conforms to the naive case where in a non emergency event patients health data could be looked at quickly to determine the likely hood of heart disease.

Question 3 Code

```
#Partitioned the data
train_index = createDataPartition(heart_Data$Target, p = 0.6, list = FALSE)

#Training data created
trainDF = heart_Data[train_index,]

#Validation data created
validationDF = heart_Data[-train_index,]

#Creation of the Naive Bayes Classifier based on BP and Chest Pain
```

```
nb_model = naive_bayes(Target ~ BP_New + chest_pain_type,
                       data = trainDF, laplace = 0)
```

```
## Warning: naive_bayes(): Feature BP_New - zero probabilities are present.
## Consider Laplace smoothing.
```

```
#Predictions made using the Classifier
train_pred = predict(nb_model, validationDF)
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```
#confusion matrix
validationDF$Target = as.factor(validationDF$Target)
confusionMatrix(train_pred, validationDF$Target, positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction NO YES
##        NO  70   0
##        YES 49   1
##
##                Accuracy : 0.5917
##                  95% CI : (0.4982, 0.6805)
##     No Information Rate : 0.9917
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0233
##
##  Mcnemar’s Test P-Value : 7.025e-12
##
##             Sensitivity : 1.000000
##             Specificity : 0.588235
##          Pos Pred Value : 0.020000
##          Neg Pred Value : 1.000000
##              Prevalence : 0.008333
##          Detection Rate : 0.008333
##    Detection Prevalence : 0.416667
##       Balanced Accuracy : 0.794118
##
##        ’Positive’ Class : YES
##
```

Question 3 Explination

After partitioning our data and training a naive Bayes classifier a confusion matrix was created which shows that we have many misclassifications. However, our sensitivity is 1 and that is good in this case because we do not want to miss anyone that has heart disease. After multiple runs the sensitivity has stayed at 1. I also tested a laplace of 1 to see if this would change any of the results due to some conditions that would have a probability of 0. However, the results were similar and the sensitivity stayed at 1.