Coronary Heart Disease Prediction

Mounika Yallamandhala

2023-04-28

Importing the data set

```
train = read.csv("D:/MS/Michigan/Courses/Business analytics problem solving ITM 883/Project/train_data (
test = read.csv("D:/MS/Michigan/Courses/Business analytics problem solving ITM 883/Project/test_data CV
train_quan_vars <- subset(train, select=c(sysBP, education, age, cigsPerDay, totChol, diaBP, BMI, heartl
round(cor(train_quan_vars, use="pairwise.complete.obs"), 2)
```

```
##             sysBP education   age cigsPerDay totChol diaBP   BMI heartRate
## sysBP        1.00     -0.14  0.40      -0.10    0.20  0.78  0.33      0.18
## education   -0.14      1.00 -0.17       0.01   -0.02 -0.06 -0.13     -0.05
## age          0.40     -0.17  1.00      -0.19    0.27  0.22  0.14      0.00
## cigsPerDay  -0.10      0.01 -0.19       1.00   -0.02 -0.07 -0.10      0.07
## totChol      0.20     -0.02  0.27      -0.02    1.00  0.15  0.11      0.09
## diaBP        0.78     -0.06  0.22      -0.07    0.15  1.00  0.38      0.17
## BMI          0.33     -0.13  0.14      -0.10    0.11  0.38  1.00      0.07
## heartRate    0.18     -0.05  0.00       0.07    0.09  0.17  0.07      1.00
## glucose      0.14     -0.04  0.12      -0.07    0.06  0.07  0.09      0.09
##             glucose
## sysBP          0.14
## education     -0.04
## age            0.12
## cigsPerDay    -0.07
## totChol        0.06
## diaBP          0.07
## BMI            0.09
## heartRate      0.09
## glucose        1.00
```

```
summary(train)
```

```
##        id              age          education         sex
##  Min.   :   0.0   Min.   :32.00   Min.   :1.000   Length:3390
##  1st Qu.: 847.2   1st Qu.:42.00   1st Qu.:1.000   Class :character
##  Median :1694.5   Median :49.00   Median :2.000   Mode  :character
##  Mean   :1694.5   Mean   :49.54   Mean   :1.971
##  3rd Qu.:2541.8   3rd Qu.:56.00   3rd Qu.:3.000
##  Max.   :3389.0   Max.   :70.00   Max.   :4.000
##                                   NA's   :87
##   is_smoking          cigsPerDay        BPMeds        prevalentStroke
```

```
##  Length:3390        Min.   : 0.000   Min.   :0.00000   Min.   :0.00000
##  Class :character   1st Qu.: 0.000   1st Qu.:0.00000   1st Qu.:0.00000
##  Mode  :character   Median : 0.000   Median :0.00000   Median :0.00000
##                     Mean   : 9.069   Mean   :0.02989   Mean   :0.00649
##                     3rd Qu.:20.000   3rd Qu.:0.00000   3rd Qu.:0.00000
##                     Max.   :70.000   Max.   :1.00000   Max.   :1.00000
##                     NA's   :22       NA's   :44
##   prevalentHyp       diabetes          totChol          sysBP
##  Min.   :0.0000   Min.   :0.00000   Min.   :107.0   Min.   : 83.5
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:206.0   1st Qu.:117.0
##  Median :0.0000   Median :0.00000   Median :234.0   Median :128.5
##  Mean   :0.3153   Mean   :0.02566   Mean   :237.1   Mean   :132.6
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:264.0   3rd Qu.:144.0
##  Max.   :1.0000   Max.   :1.00000   Max.   :696.0   Max.   :295.0
##                                     NA's   :38
##      diaBP             BMI          heartRate         glucose
##  Min.   : 48.00   Min.   :15.96   Min.   : 45.00   Min.   : 40.00
##  1st Qu.: 74.50   1st Qu.:23.02   1st Qu.: 68.00   1st Qu.: 71.00
##  Median : 82.00   Median :25.38   Median : 75.00   Median : 78.00
##  Mean   : 82.88   Mean   :25.79   Mean   : 75.98   Mean   : 82.09
##  3rd Qu.: 90.00   3rd Qu.:28.04   3rd Qu.: 83.00   3rd Qu.: 87.00
##  Max.   :142.50   Max.   :56.80   Max.   :143.00   Max.   :394.00
##                   NA's   :14      NA's   :1        NA's   :304
##    TenYearCHD
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1507
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

Removing Missing values in the data

```
missing_values = colSums(is.na(train))
print(missing_values)
```

```
##              id             age       education             sex      is_smoking
##               0               0              87               0               0
##      cigsPerDay          BPMeds prevalentStroke    prevalentHyp        diabetes
##              22              44               0               0               0
##         totChol           sysBP           diaBP             BMI       heartRate
##              38               0               0              14               1
##         glucose      TenYearCHD
##             304               0
```

```
train = na.omit(train)
sum(is.na(train))
```

```
## [1] 0
```

```
missing_values_test = colSums(is.na(test))
print(missing_values_test)
```

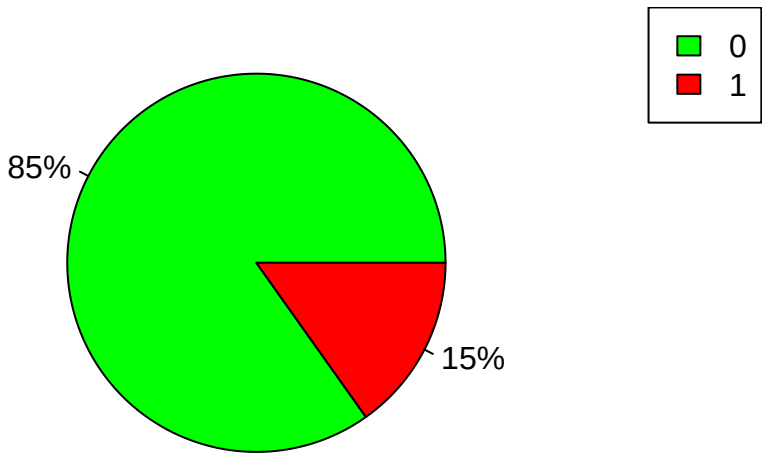
```
##              id             age       education             sex      is_smoking
##               0               0              18               0               0
##      cigsPerDay          BPMeds prevalentStroke    prevalentHyp        diabetes
##               7               9               0               0               0
##         totChol           sysBP           diaBP             BMI       heartRate
##              12               0               0               5               0
##         glucose
##              84
```

```
test = na.omit(test)
sum(is.na(test))
```

```
## [1] 0
```

Target variable distribution

```
freq = table(train$TenYearCHD)
perc = prop.table(freq)
my_colors <- c("green", "red")
my_labels <- c("0", "1")
pie(perc,labels = paste0(round(perc*100), "%"), col = my_colors, main = "TenYearCHD Binary Variable")
legend("topright", legend = my_labels, fill = my_colors)
```



Removing id column from both train and test data set

```
train = train[,-1]
test = test[,-1]
dim(train)
```

```
## [1] 2927   16
```

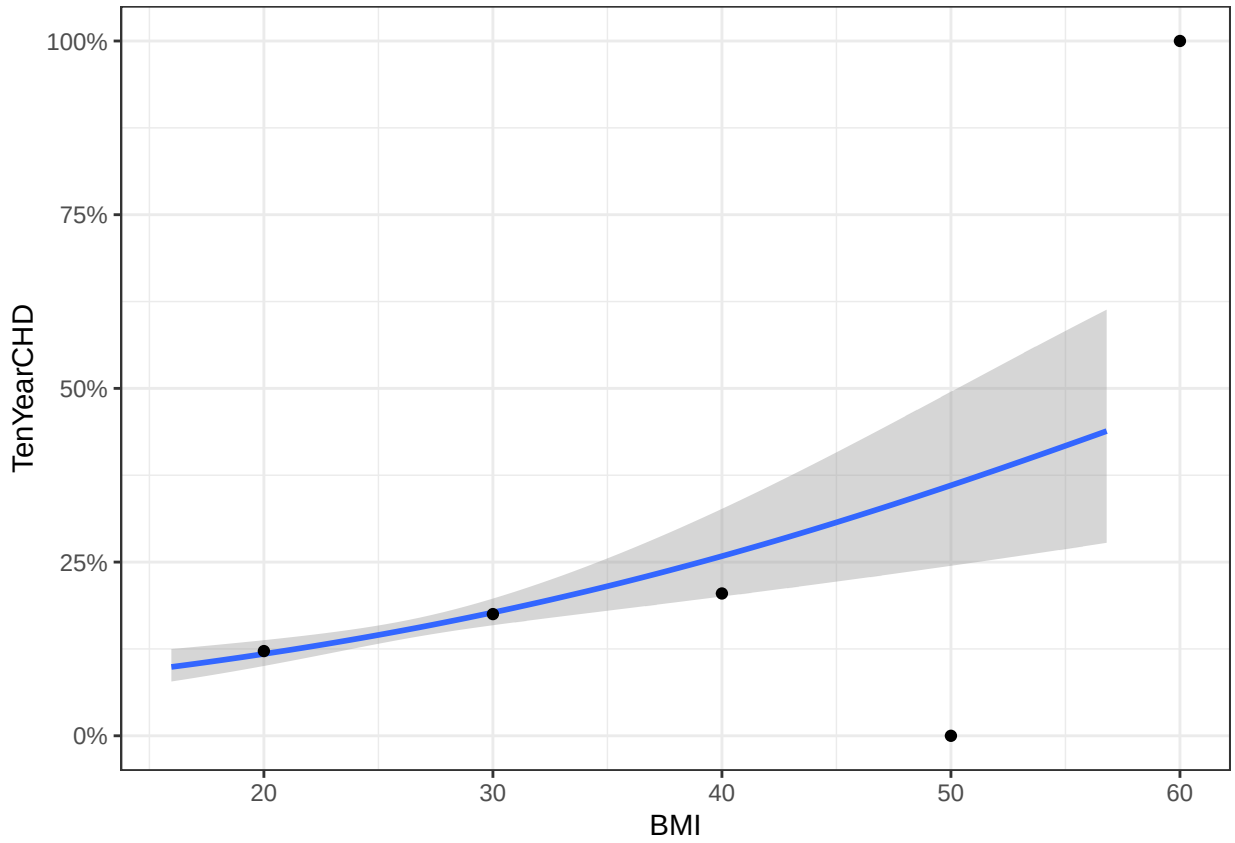
```
dim(test)
```

```
## [1] 729  15
```

Visualizing the data based on independent variables

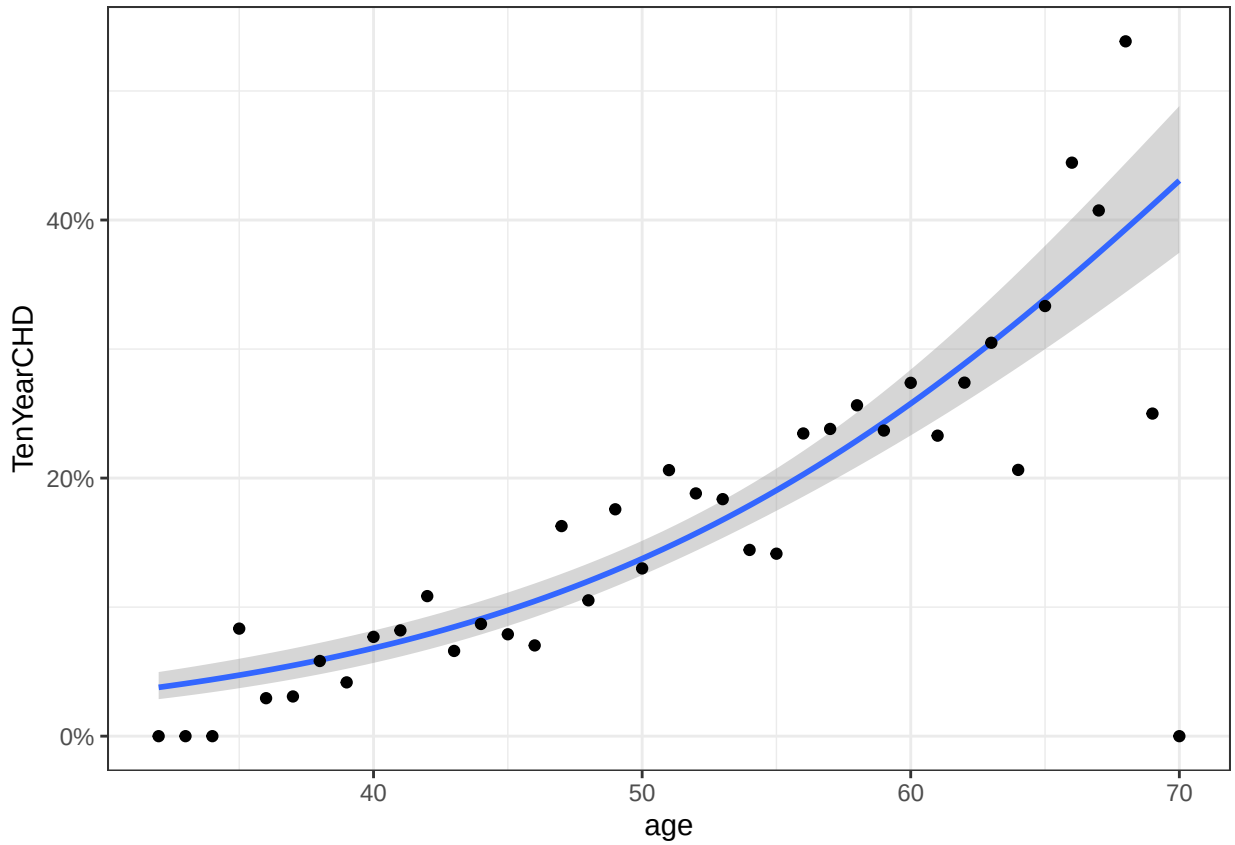
```
library(ggplot2)
ggplot(data=train) + geom_smooth(aes(x=BMI, y=TenYearCHD), method="glm",
                                 method.args = list(family = "binomial")) +
  stat_summary(data=train, aes(x=round(BMI,-1), y=TenYearCHD), fun=mean, geom="point") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data=train) + geom_smooth(aes(x=age, y=TenYearCHD), method="glm",
                                 method.args = list(family = "binomial")) +
  stat_summary(data=train, aes(x=age, y=TenYearCHD), fun=mean, geom="point") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Logistic Regression Model to predict dependent variable TenYearCHD using Age,Prevalent Stroke, Prevalent Hyp, diaBP, BMI as independent variables

```
model2 = glm(TenYearCHD ~ age + prevalentStroke + prevalentHyp + diaBP + BMI, data = train,
             family=binomial(link="logit"))
summary(model2)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ age + prevalentStroke + prevalentHyp +
##     diaBP + BMI, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3388  -0.6109  -0.4464  -0.3385   2.6076
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -6.412951   0.586124 -10.941  < 2e-16 ***
## age              0.067803   0.006692  10.132  < 2e-16 ***
## prevalentStroke  0.770105   0.511480   1.506  0.13216
```

```
## prevalentHyp     0.403090   0.138996   2.900  0.00373 **
## diaBP            0.010266   0.005399   1.901  0.05726 .
## BMI              0.006554   0.013270   0.494  0.62137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2491.6  on 2926  degrees of freedom
## Residual deviance: 2295.7  on 2921  degrees of freedom
## AIC: 2307.7
##
## Number of Fisher Scoring iterations: 5
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.2.3
```

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

```
##
## Attaching package: 'pROC'
```

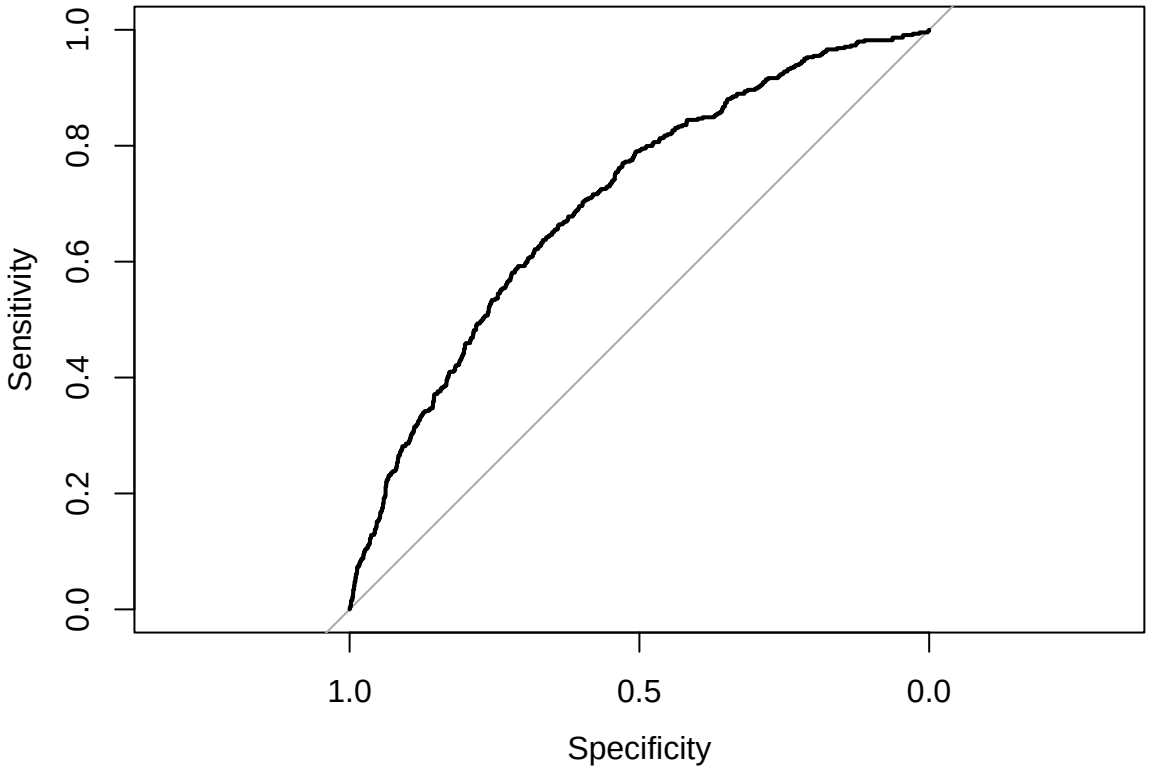
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
Predicted_prob = predict(model2, data=train, type="response")
ROC = roc(train$TenYearCHD,Predicted_prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(ROC)
```



Finding best cut-off value

```
coords(ROC,"best")
```

```
##   threshold specificity sensitivity
## 1 0.1582665   0.6665324   0.6373874
```

```
table(train$TenYearCHD)
```

```
##
##    0    1
## 2483  444
```

```
ROC
```

```
##
## Call:
## roc.default(response = train$TenYearCHD, predictor = Predicted_prob)
##
## Data: Predicted_prob in 2483 controls (train$TenYearCHD 0) < 444 cases (train$TenYearCHD 1).
## Area under the curve: 0.7027
```

Confusion matrix

```
threshold = 0.16
predicted_label <- ifelse(Predicted_prob > threshold , 1, 0)
actual_label <- train$TenYearCHD
confusion_matrix <- table(predicted_label, actual_label)
print(confusion_matrix)
```

```
##                actual_label
## predicted_label    0    1
##               0 1674  167
##               1  809  277
```

Calculate accuracy

```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.6665528
```

Calculate precision and recall

```
precision <- confusion_matrix[2,2] / sum(confusion_matrix[,2])
recall <- confusion_matrix[2,2] / sum(confusion_matrix[2,])
cat("Precision:", precision, "\n")
```

```
## Precision: 0.6238739
```

```
cat("Recall:", recall, "\n")
```

```
## Recall: 0.2550645
```