Modern Data Mining: Logistic Regression Classification and Lasso

Bopei Nie

03/24/2023

Contents

1 Part I: Framingham heart disease study

We will continue to use the Framingham Data (`Framingham.dat`) so that you are already familiar with the data and the variables. All the results are obtained through training data.

Liz is a patient with the following readings: `AGE=50, GENDER=FEMALE, SBP=110, DBP=80, CHOL=180, FRW=105, CIG=0`. We would be interested to predict Liz's outcome in heart disease.

To keep our answers consistent, use a subset of the data, and exclude anyone with a missing entry. For your convenience, we've loaded it here together with a brief summary about the data.

We note that this dataset contains 311 people diagnosed with heart disease and 1095 without heart disease.

```
   0    1
1095  311
```

After a quick cleaning up here is a summary about the data:

Lastly we would like to show five observations randomly chosen.

```
     HD AGE    SEX SBP DBP CHOL FRW CIG
643   1  61   MALE 140  68  248 104  20
11    0  45   MALE 110  88  183  90   0
576   1  58   MALE 150  95  296 100  15
560   1  59   MALE 260 130  246 111  20
702   0  45 FEMALE 122  74  178  88   5
```

1.1 Identify risk factors

1.1.1 Understand the likelihood function

Conceptual questions to understand the building blocks of logistic regression. All the codes in this part should be hidden. We will use a small subset to run a logistic regression of `HD` vs. `SBP`.

i. Take a random subsample of size 5 from `hd_data_f` which only includes `HD` and `SBP`. Also set `set.seed(471)`. List the five observations neatly below. No code should be shown here.

ii. Write down the likelihood function using the five observations above.

Since in a logistic regression model, we will model the probability of one being sick as follows:

$$P(HD = 1|SBP) = \frac{e^{\beta_0+\beta_1 SBP}}{1 + e^{\beta_0+\beta_1 SBP}}$$

where β_0 and β_1 are unknown parameters.

The maximum likelihood function is:

$$\begin{aligned}
\mathcal{L}(\beta_0, \beta_1|\text{D}ata) &= Prob(\text{the outcome of the data}) \\
&= Prob((HD = 1|SBP = 140), (HD = 0|SBP = 110), (HD = 1|SBP = 150), (HD = 1|SBP = 260), (HD = \\
&= Prob(HD = 1|SBP = 140) \times Prob(HD = 0|SBP = 110) \times Prob(HD = 1|SBP = 150) \times Prob(HD = 1|S \\
&= \frac{e^{\beta_0+140\beta_1}}{1 + e^{\beta_0+140\beta_1}} \cdot \frac{1}{1 + e^{\beta_0+110\beta_1}} \cdot \frac{e^{\beta_0+150\beta_1}}{1 + e^{\beta_0+150\beta_1}} \cdot \frac{e^{\beta_0+260\beta_1}}{1 + e^{\beta_0+260\beta_1}} \cdot \frac{1}{1 + e^{\beta_0+122\beta_1}}
\end{aligned}$$

iii. Find the MLE based on this subset using glm(). Report the estimated logit function of `SBP` and the probability of `HD`=1. Briefly explain how the MLE are obtained based on ii. above.

Using glm, we obtain the estimated logit function as follows.

The estimated logit function is:

logit = -334.96 + 2.56*SBP

That means log odds increases 2.56 when SBP increases by 1. Notice the $Prob(HD = 1)$ is an increasing function of `SBP` since $\hat{\beta}_1 = 2.56 > 0$. That means when `SBP` increases, the chance of being `HD` increases.

The probability of `HD`=1 is:

- $$\hat{P}(HD = 1|SBP) = \frac{e^{-334.96+2.56\times SBP}}{1 + e^{-334.966+2.56\times SBP}}$$

To obtain MLE based on the subset and using the glm() function, the following steps are typically taken:

Specify the model: This involves choosing the appropriate probability distribution and link function for the response variable, and specifying any covariates or predictors in the model.

Estimate the model parameters: This involves finding the parameter values that maximize the likelihood function, given the observed data. This is typically done using numerical optimization techniques.

Assess the fit of the model: This involves examining the residuals and goodness-of-fit measures to determine whether the model provides a good fit to the data.

Use the model for prediction: Once the model is fitted and validated, it can be used to make predictions on new data.

Overall, the MLE approach provides a powerful framework for fitting statistical models to data, and can be used in a wide variety of settings, including regression analysis, time series analysis, and survival analysis.

iv. Evaluate the probability of Liz having heart disease.

Based on fit1 we plug in `SBP`=110 of Liz into the prob equation.

$$\hat{P}(HD = 1|SBP = 110) = \frac{e^{-334.96+2.56\times SBP}}{1 + e^{-334.966+2.56\times SBP}} = \frac{e^{-334.96+2.56\times 110}}{1 + e^{-334.966+2.56\times 110}} \approx 0$$

We can also use the `predict()` function.

```
##        1
## 2.22e-16
```

The estimated probability of Liz having heart disease is approximately 0.

1.1.2 Identify important risk factors for `Heart.Disease`.

We focus on understanding the elements of basic inference method in this part. Let us start a fit with just one factor, `SBP`, and call it `fit1`. We then add one variable to this at a time from among the rest of the variables.

First, we obtain the regression function with one independent variable (SBP) only.

```r
fit1 <- glm(HD~SBP, hd_data.f, family=binomial)
summary(fit1)
```

```
##
## Call:
## glm(formula = HD ~ SBP, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.661  -0.709  -0.624  -0.524   2.107
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.65489    0.34787  -10.51  < 2e-16 ***
## SBP          0.01581    0.00222    7.12  1.1e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1417.5  on 1391  degrees of freedom
## AIC: 1421
##
## Number of Fisher Scoring iterations: 4
```

Then, we add different variables into the model one by one.

The model with SBP and AGE is as follow. Both variables are significant at 0.001 level.

```r
fit1.1 <- glm(HD~SBP + AGE, hd_data.f, family=binomial)
summary(fit1.1)
```

```
##
## Call:
## glm(formula = HD ~ SBP + AGE, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.630  -0.721  -0.601  -0.466   2.169
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.48625    0.79182   -8.19  2.6e-16 ***
## SBP          0.01434    0.00225    6.38  1.8e-10 ***
## AGE          0.05775    0.01422    4.06  4.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1400.8  on 1390  degrees of freedom
```

```
## AIC: 1407
##
## Number of Fisher Scoring iterations: 4
```

The model with SBP and SEX is as follow. Both variables are significant at 0.001 level.

```
fit1.2 <- glm(HD~SBP + as.factor(SEX), hd_data.f, family=binomial)
summary(fit1.2)
```

```
##
## Call:
## glm(formula = HD ~ SBP + as.factor(SEX), family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.641  -0.737  -0.573  -0.417   2.245
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -4.57026    0.38973  -11.73  < 2e-16 ***
## SBP                 0.01872    0.00232    8.05  8.1e-16 ***
## as.factor(SEX)MALE  0.90342    0.13976    6.46  1.0e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1373.8  on 1390  degrees of freedom
## AIC: 1380
##
## Number of Fisher Scoring iterations: 4
```

The model with SBP and DBP is as follow. SBP is significant at 0.001 level but DBP is not significant even at 0.05 level.

```
fit1.3 <- glm(HD~SBP + DBP, hd_data.f, family=binomial)
summary(fit1.3)
```

```
##
## Call:
## glm(formula = HD ~ SBP + DBP, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.660  -0.710  -0.623  -0.522   2.096
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.75988    0.41651   -9.03  < 2e-16 ***
## SBP          0.01449    0.00364    3.98  6.8e-05 ***
## DBP          0.00334    0.00726    0.46     0.65
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1417.3  on 1390  degrees of freedom
## AIC: 1423
##
## Number of Fisher Scoring iterations: 4
```

The model with SBP and CHOL is as follow. SBP is significant at 0.001 level and CHOL is significant at 0.05 level.

```
fit1.4 <- glm(HD~SBP + CHOL, hd_data.f, family=binomial)
summary(fit1.4)
```

```
##
## Call:
## glm(formula = HD ~ SBP + CHOL, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.734  -0.714  -0.622  -0.507   2.159
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.19172    0.46354   -9.04  < 2e-16 ***
## SBP          0.01539    0.00224    6.88  5.9e-12 ***
## CHOL         0.00254    0.00142    1.79    0.074 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1414.3  on 1390  degrees of freedom
## AIC: 1420
##
## Number of Fisher Scoring iterations: 4
```

The model with SBP and FRW is as follow. SBP is significant at 0.001 level but FRW is not significant even at 0.05 level.

```
fit1.5 <- glm(HD~SBP + FRW, hd_data.f, family=binomial)
summary(fit1.5)
```

```
##
## Call:
## glm(formula = HD ~ SBP + FRW, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
## -1.677  -0.710  -0.624  -0.523   2.110
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.74427    0.44809   -8.36  < 2e-16 ***
## SBP          0.01556    0.00236    6.61  3.9e-11 ***
## FRW          0.00120    0.00377    0.32     0.75
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1417.4  on 1390  degrees of freedom
## AIC: 1423
##
## Number of Fisher Scoring iterations: 4
```

The model with SBP and CIG is as follow. Both variables are significant at 0.001 level.

```
fit1.6 <- glm(HD~SBP + CIG, hd_data.f, family=binomial)
summary(fit1.6)
```

```
##
## Call:
## glm(formula = HD ~ SBP + CIG, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.016  -0.709  -0.604  -0.488   2.120
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.99037    0.36607  -10.90  < 2e-16 ***
## SBP          0.01687    0.00227    7.42  1.1e-13 ***
## CIG          0.02049    0.00545    3.76  0.00017 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1403.8  on 1390  degrees of freedom
## AIC: 1410
##
## Number of Fisher Scoring iterations: 4
```

i. Which single variable would be the most important to add? Add it to your model, and call the new fit `fit2`.

We will pick up the variable either with highest $|z|$ value, or smallest p value. Report the summary of your `fit2` Note: One way to keep your output neat, we will suggest you using `xtable`. And here is the summary report looks like.

From above we can see that the SEX variable has the largest $|z|$ value (6.46), so we include SEX into our model to get fit2.

```
fit2 <- glm(HD~SBP + as.factor(SEX), hd_data.f, family=binomial)
fit2
```

```
Call:  glm(formula = HD ~ SBP + as.factor(SEX), family = binomial, data = hd_data.f)

Coefficients:
       (Intercept)                SBP  as.factor(SEX)MALE
           -4.5703             0.0187              0.9034

Degrees of Freedom: 1392 Total (i.e. Null);  1390 Residual
Null Deviance:      1470
Residual Deviance: 1370     AIC: 1380
```

ii. Is the residual deviance of fit2 always smaller than that of fit1? Why or why not?

Regarding the residual deviance of fit2, it is not always smaller than that of fit1. Adding a variable to the model can increase or decrease the residual deviance, depending on whether the variable improves or worsens the fit of the model to the data. In other words, the residual deviance of fit2 depends on the specific variable added and its impact on the model.

However, in this model, SEX partially explain HD, so after adding SEX into fit2, the residual deviance decreases.

iii. Perform both the Wald test and the Likelihood ratio tests (Chi-Squared) to see if the added variable is significant at the .01 level. What are the p-values from each test? Are they the same?

Wald test is as follow:

```
summary(fit2)
```

```
##
## Call:
## glm(formula = HD ~ SBP + as.factor(SEX), family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.641  -0.737  -0.573  -0.417   2.245
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -4.57026    0.38973  -11.73  < 2e-16 ***
## SBP                 0.01872    0.00232    8.05  8.1e-16 ***
## as.factor(SEX)MALE  0.90342    0.13976    6.46  1.0e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
```

```
## Residual deviance: 1373.8  on 1390  degrees of freedom
## AIC: 1380
##
## Number of Fisher Scoring iterations: 4
```

```
confint.default(fit2)
```

```
##                       2.5 %  97.5 %
## (Intercept)        -5.3341 -3.8064
## SBP                 0.0142  0.0233
## as.factor(SEX)MALE  0.6295  1.1773
```

Similar to F tests in OLS (Ordinary Least Squares), we have likelihood ratio test to test if a collective set of variables are not needed. Likelihood ratio test is as follow:

```
anova(fit2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: HD
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                           1392       1469
## SBP             1     51.9      1391       1417  5.9e-13 ***
## as.factor(SEX)  1     43.7      1390       1374  3.8e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The tests show that it is significant at 0.01 level. p-value from Wald test is 1.0e-10, and p-value from Likelihood ratio test is 3.8e-11. They are not the same, but they are close to each other (approximately zero).

1.1.3 Model building

Start with all variables. Our goal is to fit a well-fitting model, that is still small and easy to interpret (parsimonious).

i. Use backward selection method. Only keep variables whose coefficients are significantly different from 0 at .05 level. Kick out the variable with the largest p-value first, and then re-fit the model to see if there are other variables you want to kick out.

First, put all variables into the model.

```
fit.back1 <- glm(HD~SBP + AGE + as.factor(SEX) +DBP + CHOL + FRW + CIG, hd_data.f, family=binomial)
summary(fit.back1)
```

```
##
## Call:
## glm(formula = HD ~ SBP + AGE + as.factor(SEX) + DBP + CHOL +
##     FRW + CIG, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.705  -0.727  -0.556  -0.333   2.446
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -9.33480    1.03663   -9.00  < 2e-16 ***
## SBP                 0.01484    0.00389    3.82  0.00013 ***
## AGE                 0.06249    0.01500    4.17  3.1e-05 ***
## as.factor(SEX)MALE  0.90610    0.15764    5.75  9.0e-09 ***
## DBP                 0.00288    0.00762    0.38  0.70594
## CHOL                0.00446    0.00151    2.96  0.00305 **
## FRW                 0.00580    0.00406    1.43  0.15296
## CIG                 0.01231    0.00609    2.02  0.04315 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1343.1  on 1385  degrees of freedom
## AIC: 1359
##
## Number of Fisher Scoring iterations: 4
```

From the results, we kick out DBP, which has the largest p-value.

```
fit.back2 <- glm(HD~SBP + AGE + as.factor(SEX) + CHOL + FRW + CIG, hd_data.f, family=binomial)
summary(fit.back2)
```

```
##
## Call:
## glm(formula = HD ~ SBP + AGE + as.factor(SEX) + CHOL + FRW +
##     CIG, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.707  -0.728  -0.552  -0.334   2.450
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -9.22786    0.99615   -9.26  < 2e-16 ***
## SBP                 0.01597    0.00249    6.42  1.4e-10 ***
## AGE                 0.06153    0.01478    4.16  3.1e-05 ***
## as.factor(SEX)MALE  0.91127    0.15712    5.80  6.6e-09 ***
## CHOL                0.00449    0.00150    2.99   0.0028 **
## FRW                 0.00604    0.00400    1.51   0.1315
## CIG                 0.01228    0.00609    2.02   0.0437 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1343.3  on 1386  degrees of freedom
## AIC: 1357
##
## Number of Fisher Scoring iterations: 4
```

Then, we kick out `FRW`, whose p-value is the largest in the model above.

```
fit.back3 <- glm(HD~SBP + AGE + as.factor(SEX) + CHOL + CIG, hd_data.f, family=binomial)
summary(fit.back3)
```

```
##
## Call:
## glm(formula = HD ~ SBP + AGE + as.factor(SEX) + CHOL + CIG, family = binomial,
##     data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.754  -0.729  -0.554  -0.344   2.447
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -8.70228    0.92683   -9.39  < 2e-16 ***
## SBP                 0.01709    0.00237    7.20  5.9e-13 ***
## AGE                 0.06136    0.01475    4.16  3.2e-05 ***
## as.factor(SEX)MALE  0.88575    0.15579    5.69  1.3e-08 ***
## CHOL                0.00440    0.00150    2.93   0.0033 **
## CIG                 0.01136    0.00606    1.87   0.0608 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1345.5  on 1387  degrees of freedom
## AIC: 1358
##
## Number of Fisher Scoring iterations: 4
```

Similarly, because `CIG` is not significant at 0.05 level, we eliminate it.

```
fit.back4 <- glm(HD~SBP + AGE + as.factor(SEX) + CHOL, hd_data.f, family=binomial)
summary(fit.back4)
```

```
##
## Call:
## glm(formula = HD ~ SBP + AGE + as.factor(SEX) + CHOL, family = binomial,
```

```
##     data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.607  -0.735  -0.552  -0.348   2.434
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -8.40872    0.90860   -9.25  < 2e-16 ***
## SBP                 0.01696    0.00236    7.18  7.0e-13 ***
## AGE                 0.05664    0.01450    3.91  9.4e-05 ***
## as.factor(SEX)MALE  0.98987    0.14505    6.82  8.8e-12 ***
## CHOL                0.00448    0.00150    3.00   0.0027 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1349.0  on 1388  degrees of freedom
## AIC: 1359
##
## Number of Fisher Scoring iterations: 4
```

Ultimately, we get the final model (fit.back4) as above. All variables here are significant at 0.05 level.

ii. Use AIC as the criterion for model selection. Find a model with small AIC through exhaustive search. Does exhaustive search guarantee that the p-values for all the remaining variables are less than .05? Is our final model here the same as the model from backwards elimination?

First, we prepare what we need to perform the calculation.

```
# Get the design matrix without 1's and HD
Xy_design <- model.matrix(HD ~.+0, hd_data.f)
# Attach y as the last column.
Xy <- data.frame(Xy_design, hd_data.f$HD)
fit.all <- bestglm(Xy, family = binomial, method = "exhaustive", IC="AIC", nvmax = 10)
```

```
## Morgan-Tatar search since family is non-gaussian.
```

Then, we get some of the bestmodels by exhastive search.

```
fit.all$BestModels
```

```
##    AGE SEXFEMALE SEXMALE  SBP   DBP CHOL   FRW   CIG Criterion
## 1 TRUE     FALSE    TRUE TRUE FALSE TRUE  TRUE  TRUE      1355
## 2 TRUE      TRUE   FALSE TRUE FALSE TRUE  TRUE  TRUE      1355
## 3 TRUE     FALSE    TRUE TRUE FALSE TRUE FALSE  TRUE      1356
## 4 TRUE      TRUE   FALSE TRUE FALSE TRUE FALSE  TRUE      1356
## 5 TRUE     FALSE    TRUE TRUE FALSE TRUE FALSE FALSE      1357
```

```
fit.all$BestModel
```

```
##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
## (Intercept)          AGE      SEXMALE          SBP         CHOL          FRW
##    -9.22786      0.06153      0.91127      0.01597      0.00449      0.00604
##         CIG
##     0.01228
##
## Degrees of Freedom: 1392 Total (i.e. Null);  1386 Residual
## Null Deviance:       1470
## Residual Deviance: 1340  AIC: 1360
```

After that, we check if all variables are significant at 0.05 level.

```
summary(glm(HD~AGE+as.factor(SEX)+SBP+CHOL+FRW+CIG, family=binomial, data=hd_data.f))
```

```
##
## Call:
## glm(formula = HD ~ AGE + as.factor(SEX) + SBP + CHOL + FRW +
##     CIG, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.707  -0.728  -0.552  -0.334   2.450
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -9.22786    0.99615   -9.26  < 2e-16 ***
## AGE                 0.06153    0.01478    4.16  3.1e-05 ***
## as.factor(SEX)MALE  0.91127    0.15712    5.80  6.6e-09 ***
## SBP                 0.01597    0.00249    6.42  1.4e-10 ***
## CHOL                0.00449    0.00150    2.99   0.0028 **
## FRW                 0.00604    0.00400    1.51   0.1315
## CIG                 0.01228    0.00609    2.02   0.0437 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1343.3  on 1386  degrees of freedom
## AIC: 1357
##
## Number of Fisher Scoring iterations: 4
```

From the results, we conclude that exhaustive search does not guarantee that the p-values for all the remaining variables are less than 0.05.

Since FRW is not significant at 0.05 level, we should eliminate it.

```r
fit.then <- glm(HD~AGE+as.factor(SEX)+SBP+CHOL+CIG, family=binomial, data=hd_data.f)
summary(fit.then)
```

```
##
## Call:
## glm(formula = HD ~ AGE + as.factor(SEX) + SBP + CHOL + CIG, family = binomial,
##     data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.754  -0.729  -0.554  -0.344   2.447
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -8.70228    0.92683   -9.39  < 2e-16 ***
## AGE                 0.06136    0.01475    4.16  3.2e-05 ***
## as.factor(SEX)MALE  0.88575    0.15579    5.69  1.3e-08 ***
## SBP                 0.01709    0.00237    7.20  5.9e-13 ***
## CHOL                0.00440    0.00150    2.93   0.0033 **
## CIG                 0.01136    0.00606    1.87   0.0608 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1345.5  on 1387  degrees of freedom
## AIC: 1358
##
## Number of Fisher Scoring iterations: 4
```

```r
fit.final <- glm(HD~AGE+as.factor(SEX)+SBP+CHOL, family=binomial, data=hd_data.f)
summary(fit.final)
```

```
##
## Call:
## glm(formula = HD ~ AGE + as.factor(SEX) + SBP + CHOL, family = binomial,
##     data = hd_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.607  -0.735  -0.552  -0.348   2.434
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -8.40872    0.90860   -9.25  < 2e-16 ***
## AGE                 0.05664    0.01450    3.91  9.4e-05 ***
## as.factor(SEX)MALE  0.98987    0.14505    6.82  8.8e-12 ***
## SBP                 0.01696    0.00236    7.18  7.0e-13 ***
## CHOL                0.00448    0.00150    3.00   0.0027 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1349.0  on 1388  degrees of freedom
## AIC: 1359
##
## Number of Fisher Scoring iterations: 4
```

The model obtained by exhaustive search is different from the model we get from backward selection, because exhaustive search model contains `FRW` and `CIG` additionally. However, after eliminating the insignificant varibales, the final model we get is the same as the model obtained from backwards elimination, since `FRW` and `CIG` are eliminated.

iii. Use the model chosen from part ii. as the final model. Write a brief summary to describe important factors relating to Heart Diseases (i.e. the relationships between those variables in the model and heart disease). Give a definition of "important factors".

Based on the final model obtained in ii, we may say collectively `AGE`, `SBP`, `SEX`, `CHOL` are all positively related to the chance of a `HD`. Those factors are important at 0.05 level.

Important factors are those significant at 0.05 level, including `SBP`, `SEX`, `AGE` and `CHOL`.

To be more precise, as `AGE`, `SBP`, `CHOL` increases, the estimated probability of having `HD` increases. Also, male's have higher chance of `HD` comparing with females controlling for all other factors in the model.

iv. What is the probability that Liz will have heart disease, according to our final model?

```
df <- data.frame(HD = c(NA), AGE=50, SEX='FEMALE', SBP=110, CHOL=180)
fit.final.predict <- predict(fit.final, df, type="response")
fit.final.predict
```

```
##      1
## 0.0519
```

Since Liz is a patient with the following readings: AGE=50, GENDER=FEMALE, SBP=110, DBP=80, CHOL=180, FRW=105, CIG=0.

According to our final model, the probability that Liz will have heart disease is estimated to be 0.0519. The result indicates a very low probability of having heart disease.

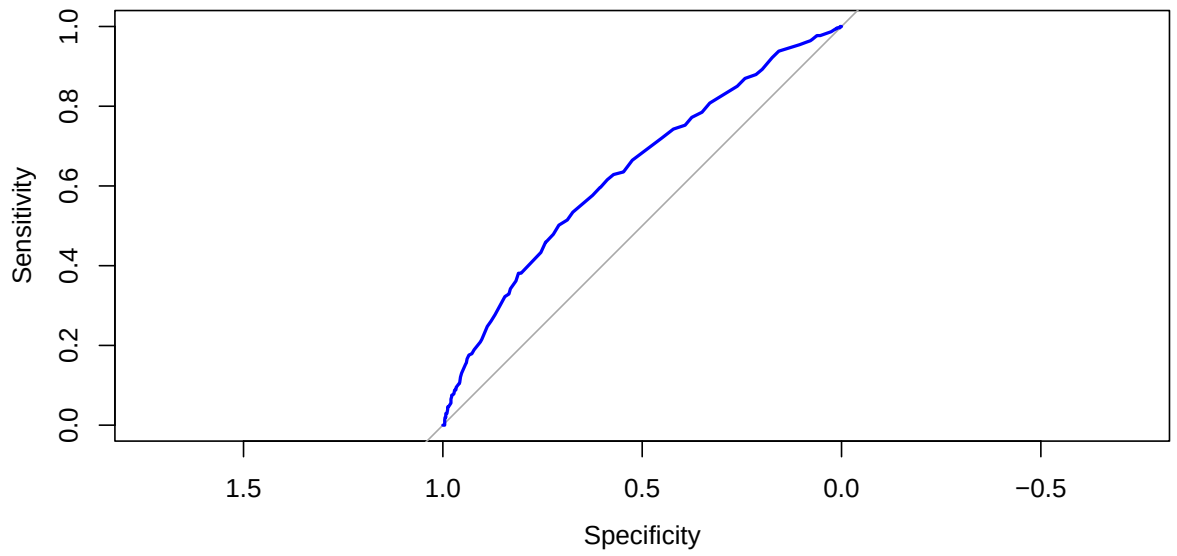
1.2 Classification analysis

1.2.1 ROC/FDR

i. Display the ROC curve using `fit1`. Explain what ROC reports and how to use the graph. Specify the classifier such that the False Positive rate is less than .1 and the True Positive rate is as high as possible.

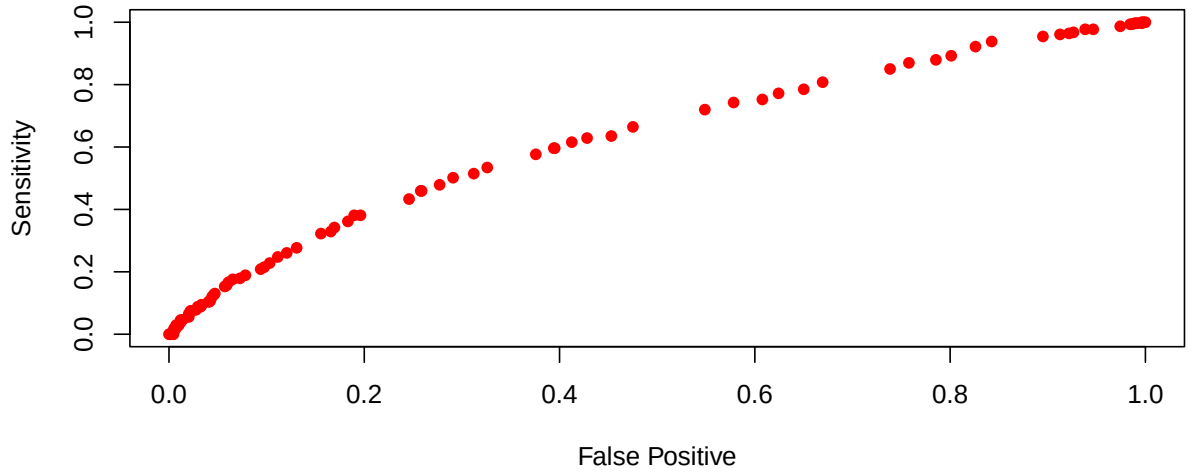
```
library(pROC)

fit1.roc <- roc(hd_data.f$HD, fit1$fitted, plot=T, col="blue")
```

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

## Setting direction: controls < cases
```



ROC curve here is Sensitivity (TPR) vs. Specificity (FPR). ROC curve is helpful when choosing classifiers. We want to have both high specificity and high sensitivity at the same time, which means we want to classify both Y = 0 and Y = 1 correctly. However, in general, we will NOT have a perfect classifier and need to strive a balance between the two.

```
plot(1-fit1.roc$specificities, fit1.roc$sensitivities, col="red", pch=16, xlab="False Positive", ylab="
```



```r
#FPR
1-fit1.roc$specificities
```

```
##  [1] 1.000000 0.998158 0.996317 0.996317 0.991713 0.990792 0.988950 0.986188
##  [9] 0.984346 0.974217 0.946593 0.938306 0.926335 0.921731 0.912523 0.895028
## [17] 0.842541 0.825967 0.801105 0.785451 0.757827 0.738490 0.669429 0.650092
## [25] 0.624309 0.607735 0.578269 0.548803 0.475138 0.453039 0.428177 0.412523
## [33] 0.395028 0.394107 0.375691 0.325967 0.312155 0.290976 0.277164 0.258748
## [41] 0.257827 0.245856 0.196133 0.189687 0.183241 0.169429 0.165746 0.155617
## [49] 0.130755 0.120626 0.111418 0.103131 0.097606 0.093923 0.078269 0.072744
## [57] 0.065378 0.060773 0.058932 0.057090 0.046961 0.046041 0.044199 0.042357
## [65] 0.041436 0.040516 0.033149 0.033149 0.032228 0.030387 0.029466 0.027624
## [73] 0.022099 0.022099 0.021179 0.020258 0.020258 0.013812 0.011971 0.011971
## [81] 0.011971 0.010129 0.009208 0.007366 0.007366 0.005525 0.004604 0.004604
## [89] 0.004604 0.004604 0.003683 0.002762 0.001842 0.000921 0.000000
```

```r
1-fit1.roc$specificities[53]
```

```
## [1] 0.0976
```

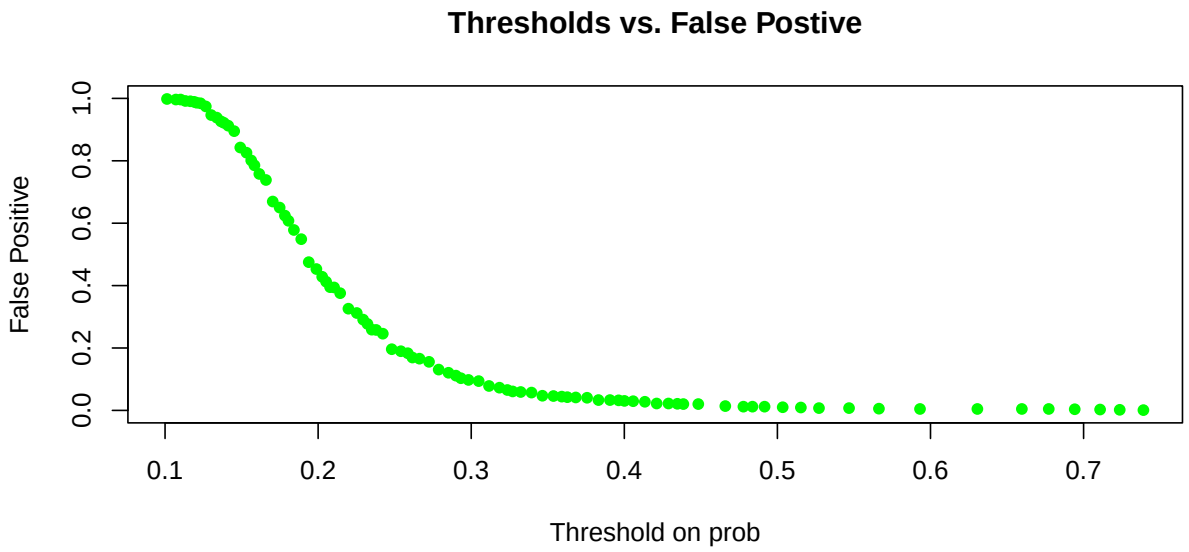
```r
#TPR
fit1.roc$sensitivities
```

```
##  [1] 1.00000 1.00000 1.00000 0.99674 0.99674 0.99674 0.99674 0.99349 0.99349
## [10] 0.98697 0.97720 0.97720 0.96743 0.96417 0.96091 0.95440 0.93811 0.92182
## [19] 0.89251 0.87948 0.86971 0.85016 0.80782 0.78502 0.77199 0.75244 0.74267
## [28] 0.71987 0.66450 0.63518 0.62866 0.61564 0.59609 0.59609 0.57655 0.53420
## [37] 0.51466 0.50163 0.47883 0.45928 0.45928 0.43322 0.38111 0.38111 0.36156
## [46] 0.34202 0.32899 0.32248 0.27687 0.26059 0.24756 0.22801 0.21498 0.20847
## [55] 0.18893 0.17915 0.17590 0.16612 0.15635 0.15309 0.13029 0.12704 0.12052
## [64] 0.10749 0.10423 0.10423 0.09446 0.09121 0.08795 0.08795 0.08795 0.07818
## [73] 0.07492 0.07166 0.06840 0.06515 0.05537 0.04560 0.04560 0.04235 0.03909
## [82] 0.02932 0.02932 0.02932 0.01954 0.01954 0.01303 0.00651 0.00326 0.00000
## [91] 0.00000 0.00000 0.00000 0.00000 0.00000
```

```r
fit1.roc$sensitivities[53]
```

```
## [1] 0.215
```

```r
plot(fit1.roc$thresholds, 1-fit1.roc$specificities, col="green", pch=16,
xlab="Threshold on prob",
ylab="False Positive",
main = "Thresholds vs. False Postive")
```




```
fit1.roc$thresholds[53]
```

```
## [1] 0.298
```

The classifier with the False Positive rate less than .1 and the True Positive rate as high as possible is when the FPR=0.0976 and TPR=0.215. The threshold on probability is 0.298, so HD = 1 if prob > 0.298.

ii. Overlay two ROC curves: one from `fit1`, the other from `fit2`. Does one curve always contain the other curve? Is the AUC of one curve always larger than the AUC of the other one? Why or why not?

```
fit1.roc <- roc(hd_data.f$HD, fit1$fitted)
```

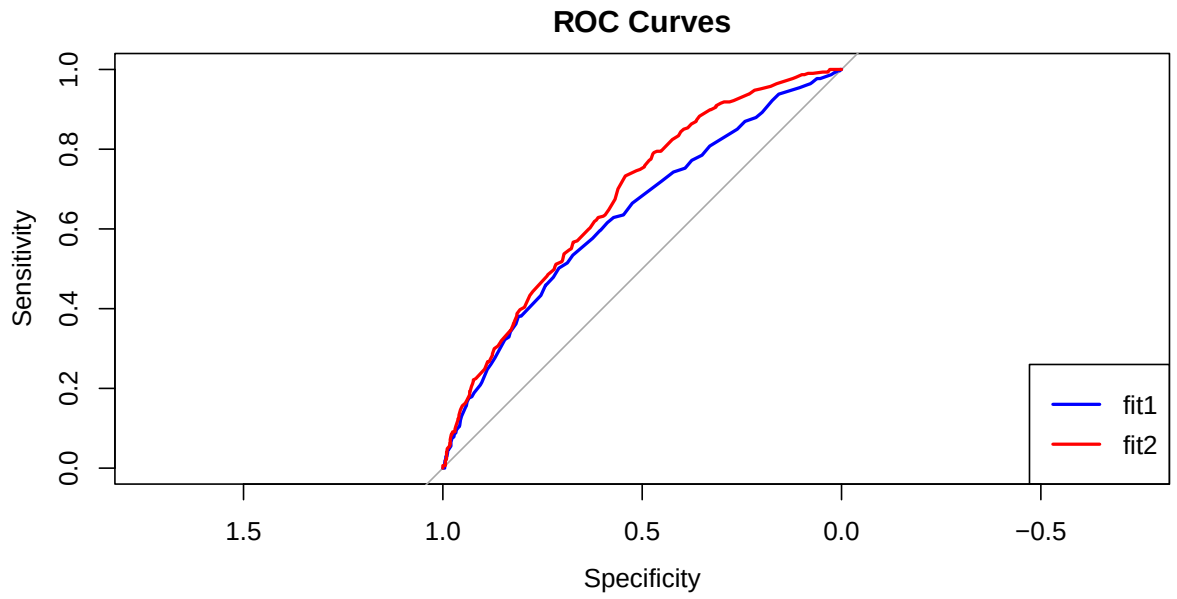
```
## Setting levels: control = 0, case = 1
```

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

```
fit2.roc <- roc(hd_data.f$HD, fit2$fitted)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(fit1.roc, main = "ROC Curves", col = "blue")
lines(fit2.roc, col = "red")
legend("bottomright", legend = c("fit1", "fit2"), col = c("blue", "red"), lwd = 2)
```



fit2's roc curve contains fit1's roc curve.

```
fit1.roc$auc
pROC::auc(fit1.roc)
```

```
## Area under the curve: 0.636
## Area under the curve: 0.636
```

```
fit2.roc$auc
pROC::auc(fit2.roc)
```

```
## Area under the curve: 0.68
## Area under the curve: 0.68
```

The AUC of fit2, 0.68, is larger than the AUC of fit1, which is 0.636. This means fit2 performs better as a classification model.

iii. Estimate the Positive Prediction Values and Negative Prediction Values for `fit1` and `fit2` using .5 as a threshold. Which model is more desirable if we prioritize the Positive Prediction values?

```
fit1.pred <- ifelse(fit1$fitted > 0.5, "1", "0")
cm1 <- table(fit1.pred, hd_data.f$HD)
cm1
```

```
##
## fit1.pred    0    1
##         0 1075  298
##         1   11    9
```

```
positive.pred <- cm1[2,2] / sum(cm1[2,])
positive.pred
```

```
## [1] 0.45
```

```
negative.pred <- cm1[1,1] / sum(cm1[1,])
negative.pred
```

```
## [1] 0.783
```

```
fit2.pred <- ifelse(fit2$fitted > 0.5, "1", "0")
cm2 <- table(fit2.pred, hd_data.f$HD)
cm2
```

```
##
## fit2.pred    0    1
##         0 1067  290
##         1   19   17
```

```
positive.pred <- cm2[2,2] / sum(cm2[2,])
positive.pred
```

```
## [1] 0.472
```

```
negative.pred <- cm2[1,1] / sum(cm2[1,])
negative.pred
```

```
## [1] 0.786
```

Positive Prediction Values is 0.45 for fit1 and 0.472 for fit2. Negative Prediction Values is 0.783 for fit1 and 0.786 for fit2. If we prioritize the Positive Prediction values, fit2 has a larger value, so we prefer fit2.

iv. For `fit1`: overlay two curves, but put the threshold over the probability function as the x-axis and positive prediction values and the negative prediction values as the y-axis. Overlay the same plot for `fit2`. Which model would you choose if the set of positive and negative prediction values are the concerns? If you can find an R package to do so, you may use it directly.

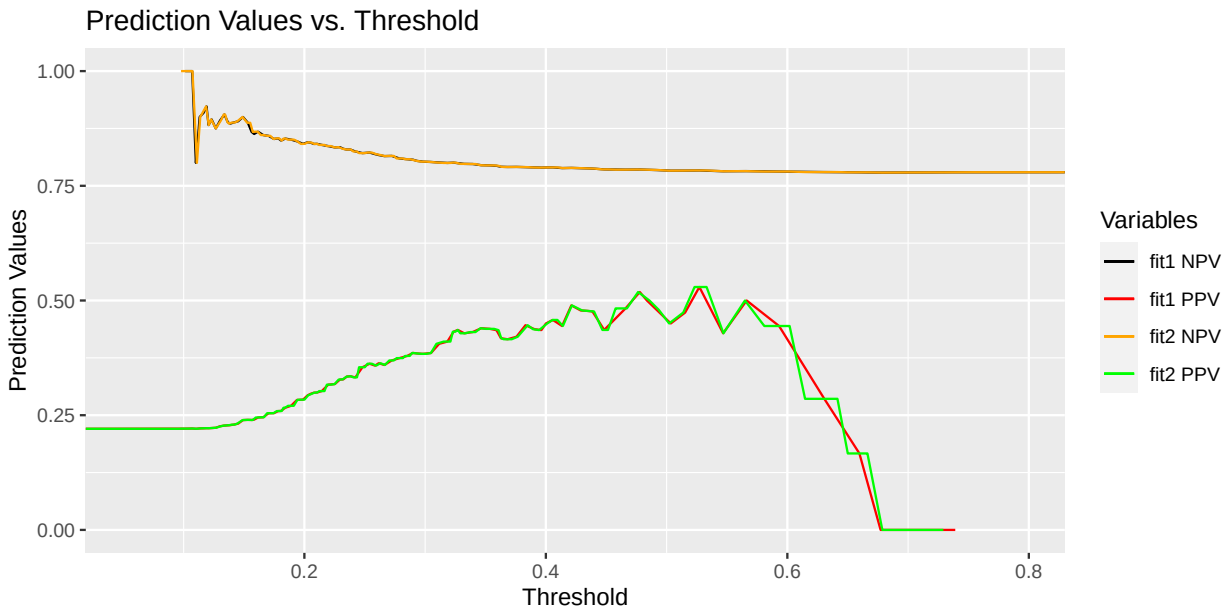
```
library(ggplot2)

x1 <- fit1.roc$thresholds

y1 <- numeric(length(x1))
for (i in seq_along(x1)) {
  pred_labels <- ifelse(fit1$fitted > x1[i], "1", "0")
  TP <- sum(pred_labels == "1" & hd_data.f$HD == "1")
  PP <- sum(pred_labels == "1")
  y1[i] <- TP / PP
}
```

```r
y2 <- numeric(length(x1))
for (i in seq_along(x1)) {
  pred_labels <- ifelse(fit1$fitted > x1[i], "1", "0")
  TP <- sum(pred_labels == "0" & hd_data.f$HD == "0")
  PP <- sum(pred_labels == "0")
  y2[i] <- TP / PP
}

df1 <- data.frame(x1, y1, y2)

x2 <- fit2.roc$thresholds

y3 <- numeric(length(x2))
for (i in seq_along(x2)) {
  pred_labels <- ifelse(fit1$fitted > x2[i], "1", "0")
  TP <- sum(pred_labels == "1" & hd_data.f$HD == "1")
  PP <- sum(pred_labels == "1")
  y3[i] <- TP / PP
}

y4 <- numeric(length(x2))
for (i in seq_along(x2)) {
  pred_labels <- ifelse(fit1$fitted > x2[i], "1", "0")
  TP <- sum(pred_labels == "0" & hd_data.f$HD == "0")
  PP <- sum(pred_labels == "0")
  y4[i] <- TP / PP
}

df2 <- data.frame(x2, y3, y4)

ggplot() +
  geom_line(data = df1, aes(x = x1, y = y1, color = "fit1 PPV")) +
  geom_line(data = df1, aes(x = x1, y = y2, color = "fit1 NPV")) +
  geom_line(data = df2, aes(x = x2, y = y3, color = "fit2 PPV")) +
  geom_line(data = df2, aes(x = x2, y = y4, color = "fit2 NPV")) +
  scale_color_manual(name = "Variables", values = c("fit1 PPV" = "red", "fit1 NPV" = "black", "fit2 PPV
  labs(x = "Threshold", y = "Prediction Values", title = "Prediction Values vs. Threshold")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
## Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 3 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 18 rows containing missing values (`geom_line()`).
```

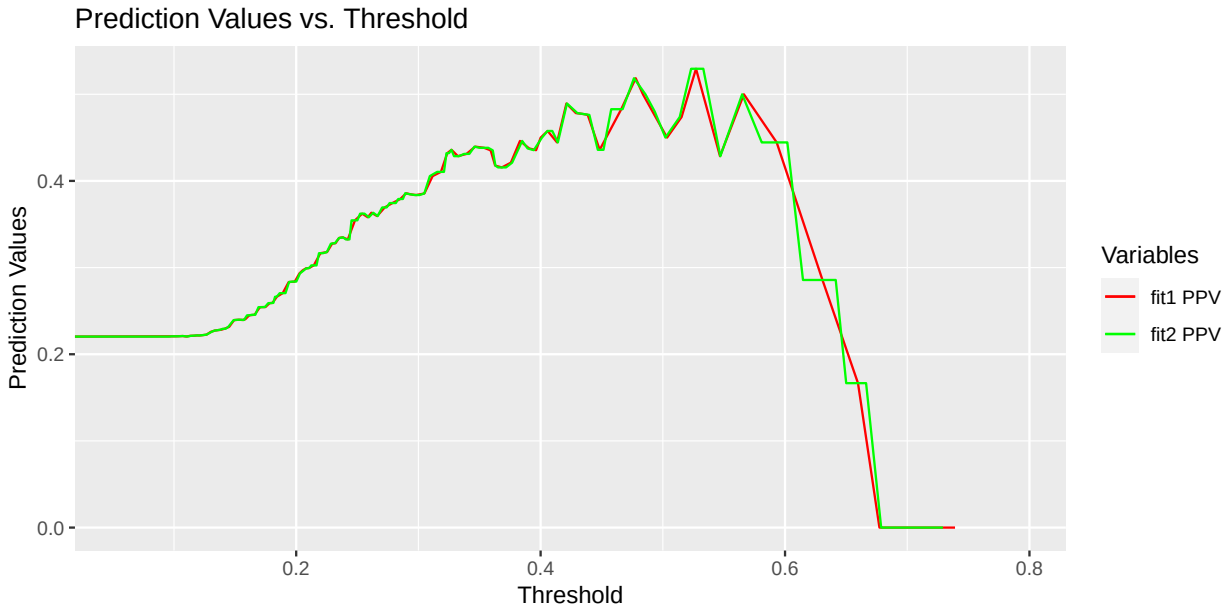


We then draw two additional graphs to help us see the differences better.

```
ggplot() +
  geom_line(data = df1, aes(x = x1, y = y1, color = "fit1 PPV")) +
  geom_line(data = df2, aes(x = x2, y = y3, color = "fit2 PPV")) +
  scale_color_manual(name = "Variables", values = c("fit1 PPV" = "red", "fit2 PPV" = "green")) +
  labs(x = "Threshold", y = "Prediction Values", title = "Prediction Values vs. Threshold")
```

```
## Warning: Removed 1 row containing missing values (‘geom_line()‘).
```

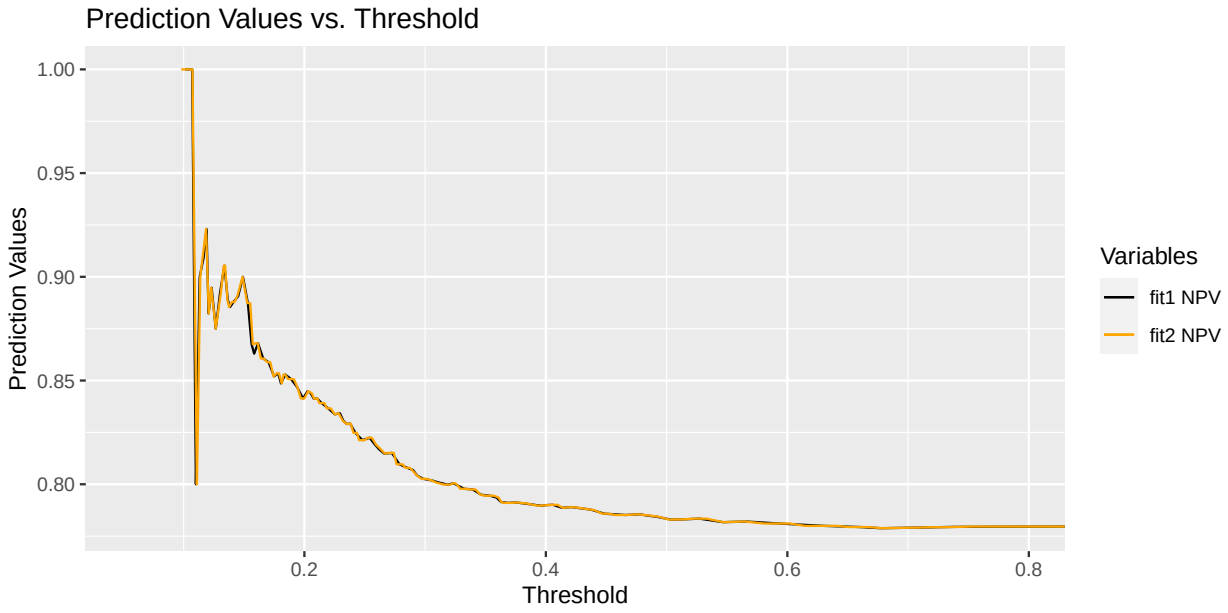
```
## Warning: Removed 3 rows containing missing values (‘geom_line()‘).
```



```
ggplot() +
  geom_line(data = df1, aes(x = x1, y = y2, color = "fit1 NPV")) +
  geom_line(data = df2, aes(x = x2, y = y4, color = "fit2 NPV")) +
  scale_color_manual(name = "Variables", values = c("fit1 NPV" = "black", "fit2 NPV" = "orange")) +
  labs(x = "Threshold", y = "Prediction Values", title = "Prediction Values vs. Threshold")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 18 rows containing missing values (`geom_line()`).
```



Considering the set of positive and negative prediction values, fit1 and fit2 have similar prediction values. fit2 has slightly higher prediction values for both positive and negative prediction values. So we favor fit2.

1.2.2 Cost function/ Bayes Rule

Bayes rules with risk ratio $\frac{a_{10}}{a_{01}} = 10$ or $\frac{a_{10}}{a_{01}} = 1$. Use your final model obtained from Part 1 to build a class of linear classifiers.

i. Write down the linear boundary for the Bayes classifier if the risk ratio of $a_{10}/a_{01} = 10$.

$a_{10} = 10a_{01}$

P_hat(Y = 1|x) > 0.1/(1+0.1) = 0.0909

logit > log(0.0909/0.9091) = -2.3027

HD_hat=1 if 0<logit+2.3027 0<0.061AGE+0.886MALE+0.017SBP+0.0044CHOL+0.011CIG-8.702+2.3027
0<0.061AGE+0.886MALE+0.017SBP+0.0044CHOL+0.011CIG-6.399

ii. What is your estimated weighted misclassification error for this given risk ratio?

```
fit.final.pred.bayes <- as.factor(ifelse(fit.final$fitted > .0909, "1", "0"))
MCE.bayes <- (5*sum(fit.final.pred.bayes[hd_data.f$HD == "1"] != "1") + sum(fit.final.pred.bayes[hd_data
MCE.bayes
```

```
## [1] 0.69
```

iii. How would you classify Liz under this classifier?

Liz is a patient with the following readings: AGE=50, GENDER=FEMALE, SBP=110, DBP=80, CHOL=180, FRW=105, CIG=0. 0.061(50)+0.886(0)+0.017(110)+0.0044(180)+0.011(0)-6.399 = -0.687 < 0. So HD_hat=0. Liz is predicted that she does not have heart diseases.

iv. Bayes rule gives us the best rule if we can estimate the probability of `HD-1` accurately. In practice we use logistic regression as our working model. How well does the Bayes rule work in practice? We hope to show in this example it works pretty well.

For the heart disease classification, falsely diagnoses as disease-free is worse than falsely diagnosed as positive. We want to prioritize lower false negative, so we weigh more on false negative. Using Bayes rules with risk ratio a10/a01=n and n>1, we can achieve putting more weight on correctly predicting HD-1.

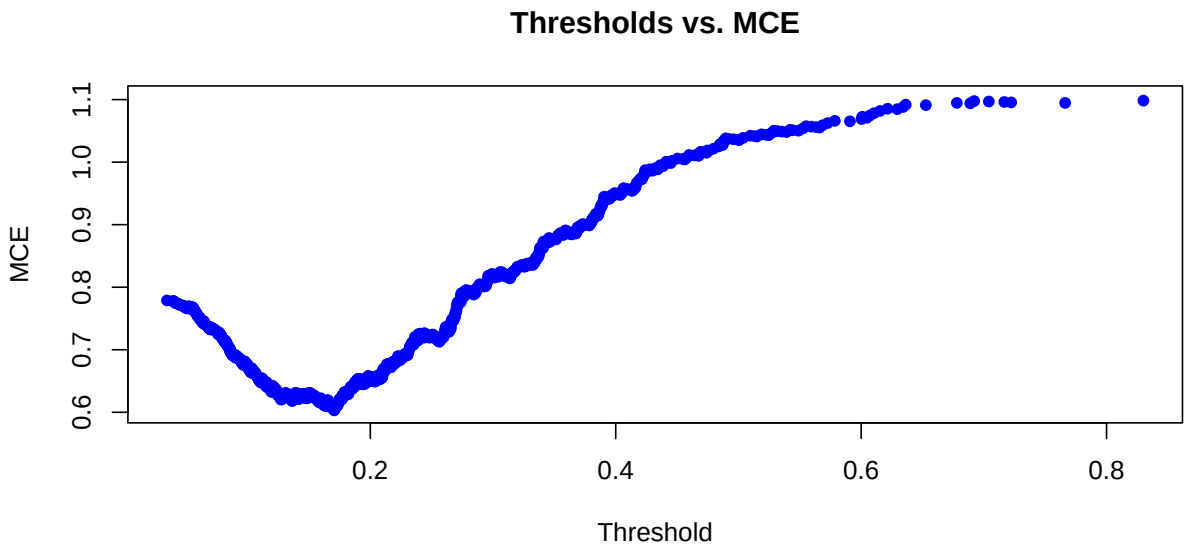
Now, draw two estimated curves where x = threshold, and y = misclassification errors, corresponding to the thresholding rule given in x-axis.

```
fit.final.roc <- roc(hd_data.f$HD, fit.final$fitted)
```

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

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

```
x<-fit.final.roc$thresholds

y <- numeric(length(x))

for (i in seq_along(x)) {
  fit.final.pred.bayes <- as.factor(ifelse(fit.final$fitted > x[i], "1", "0"))
  top <- (5*sum(fit.final.pred.bayes[hd_data.f$HD == "1"] != "1") + sum(fit.final.pred.bayes[hd_data.f$
  bottom <- length(hd_data.f$HD)
  y[i] <- top / bottom
}

plot(x, y, col="blue", pch=16,xlab="Threshold",ylab="MCE",main = "Thresholds vs. MCE")
```



v. Use weighted misclassification error, and set $a_{10}/a_{01} = 10$. How well does the Bayes rule classifier perform?

$a_{10} = 10a_{01}$ P_hat(Y = 1|x) > 0.1/(1+0.1) = 0.0909 According to the graph, MCE now is around 0.65. A smaller MCE means a better classifier. The MCE is quite small, so the Bayes rule classifier perform quite well.

vi. Use weighted misclassification error, and set $a_{10}/a_{01} = 1$. How well does the Bayes rule classifier perform?

$a_{10} = a_{01}$ P_hat(Y = 1|x) > 1/2 = 0.5 According to the graph, MCE now is around 1.0. Here we treat a10 and a01 the same. Consequently, the MCE is much larger than the previous question, so the Bayes rule classifier perform not so well.

2 Part II: Project: Lending Club Analysis

Please refer to Credit Risk via Lending Club.Rmd.