Modern Data Mining - HW 3

Kexin Zhu
Yang Yi
Yifan Jiang

Overview / Instructions

This is homework #3 of STAT 471/571/701. It will be **due on 17, March, 2019 by 11:59 PM** on Canvas. You can directly edit this file to add your answers. Submit the Rmd file, a PDF or word or HTML version with **only 1 submission** per HW team. No zip files please.

Note: To minimize your work and errors, we provide this Rmd file to guide you in the process of building your final report. To that end, we've included code to load the necessary data files. Make sure that the following files are in the same folder as this R Markdown file:

- `FRAMINGHAM.dat`
- `Bills.subset.csv`
- `Bills.subset.test.csv`

The data should load properly if you are working in Rstudio, *without needing to change your working directory.*

R Markdown / Knitr tips

You should think of this R Markdown file as generating a polished report, one that you would be happy to show other people (or your boss). There shouldn't be any extraneous output; all graphs and code run should clearly have a reason to be run. That means that any output in the final file should have explanations.

A few tips:

- Keep each chunk to only output one thing! In R, if you're not doing an assignment (with the `<-` operator), it's probably going to print something.
- If you don't want to print the R code you wrote (but want to run it, and want to show the results), use a chunk declaration like this: `{r, echo=F}`
- If you don't want to show the results of the R code or the original code, use a chunk declaration like: `{r, include=F}`
- If you don't want to show the results, but show the original code, use a chunk declaration like: `{r, results='hide'}`.
- If you don't want to run the R code at all use `{r, eval = F}`.
- We show a few examples of these options in the below example code.
- For more details about these R Markdown options, see the documentation.
- Delete the instructions and this R Markdown section, since they're not part of your overall report.

Problem 0

Review the code and concepts covered during lecture, in particular, logistic regression and classification.

Problem 1

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.

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:

```
# using the comment="     ", we get rid of the ## in the output.
summary(hd_data.f)
```

```
       HD            AGE             SEX            SBP             DBP        
       0:1086   Min.   :45.00   FEMALE:730   Min.   : 90.0   Min.   : 50.00  
       1: 307   1st Qu.:48.00   MALE  :663   1st Qu.:130.0   1st Qu.: 80.00  
                Median :52.00                Median :142.0   Median : 90.00  
                Mean   :52.43                Mean   :148.1   Mean   : 90.16  
                3rd Qu.:56.00                3rd Qu.:160.0   3rd Qu.: 98.00  
                Max.   :62.00                Max.   :300.0   Max.   :160.00  
            CHOL            FRW             CIG        
       Min.   : 96.0   Min.   : 52.0   Min.   : 0.000  
       1st Qu.:200.0   1st Qu.: 94.0   1st Qu.: 0.000  
       Median :230.0   Median :103.0   Median : 0.000  
       Mean   :234.6   Mean   :105.4   Mean   : 8.035  
       3rd Qu.:264.0   3rd Qu.:114.0   3rd Qu.:20.000  
       Max.   :430.0   Max.   :222.0   Max.   :60.000  
```

Part 1A

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(50)`. List the five observations neatly below. No code should be shown here.

```
##      HD SBP
## 996   0 142
## 614   0 126
## 281   0 136
## 1075  0 178
## 719   0 126
```

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

$$\begin{aligned}\mathcal{L}\rangle\|(\beta_0,\beta_1|Data) &= \text{Prob(the outcome of the data)}\\ &= Prob((Y=0|SBP=142),(Y=0|SBP=126),(Y=0|SBP=136),(Y=0|SBP=178),(Y=0|SBP=\\ &= \frac{1}{1+e^{\beta_0+142\beta_1}}\cdot\frac{1}{1+e^{\beta_0+126\beta_1}}\cdot\frac{1}{1+e^{\beta_0+136\beta_1}}\cdot\frac{1}{1+e^{\beta_0+178\beta_1}}\cdot\frac{1}{1+e^{\beta_0+126\beta_1}}\end{aligned}$$

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.

```
fit <- glm(HD~SBP, sample, family=binomial(logit))
summary(fit)
```

```
##
## Call:
## glm(formula = HD ~ SBP, family = binomial(logit), data = sample)
##
## Deviance Residuals:
##        996          614          281         1075          719
## -6.547e-06  -6.547e-06  -6.547e-06  -6.547e-06  -6.547e-06
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -24.57  436053.61       0        1
## SBP              0.00    3051.55       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 0.0000e+00  on 4  degrees of freedom
## Residual deviance: 2.1434e-10  on 3  degrees of freedom
## AIC: 4
##
## Number of Fisher Scoring iterations: 23
```

Thus, we have:

- logit = -24.57 + 0.00 SBP
- $P(HD = 1|SBP) = \frac{e^{-24.57+0.00\times SBP}}{1+e^{-24.57+0.00\times SBP}}$

MLE(Max Likelihood Estimator) are obtained by the likelihood function, which requires multipling the max possibility of `HD = 0` or `HD = 1` based on specific conditions that lead to corresponding results. The possibility is estimated by calculating the odds of success in or failure in terms of heart attack.

Part 1B

Goal: Identify important risk factors for `Heart.Disease.` through logistic regression. Start a fit with just one factor, `SBP`, and call it `fit1`. Let us add one variable to this at a time from among the rest of the variables.

```r
fit1 <- glm(HD~SBP, hd_data.f, family=binomial)
summary(fit1)
fit1.1 <- glm(HD~SBP + AGE, hd_data.f, family=binomial)
summary(fit1.1)
fit1.2 <- glm(HD~SBP + SEX, hd_data.f, family=binomial)
summary(fit1.2)
fit1.3 <- glm(HD~SBP + DBP, hd_data.f, family=binomial)
summary(fit1.3)
fit1.4 <- glm(HD~SBP + CHOL, hd_data.f, family=binomial)
summary(fit1.4)
fit1.5 <- glm(HD~SBP + DBP, hd_data.f, family=binomial)
summary(fit1.5)
fit1.6 <- glm(HD~SBP + FRW, hd_data.f, family=binomial)
summary(fit1.6)
fit1.7 <- glm(HD~SBP + CIG, hd_data.f, family=binomial)
summary(fit1.7)
```

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

Answer: Since we observed that fit1.2 output the smallest `AIC`, we conclude that `SEX` would be the most important to add in the model with two predictors.

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

```
##
## Call:
## glm(formula = HD ~ SBP + SEX, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6408  -0.7373  -0.5726  -0.4169   2.2452
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.570256   0.389727 -11.727  < 2e-16 ***
## SBP          0.018717   0.002324   8.053 8.07e-16 ***
## SEXMALE      0.903420   0.139762   6.464 1.02e-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: 1379.8
##
## Number of Fisher Scoring iterations: 4
```

We will pick up the variable either with highest $|z|$ value, or smallest p value. From all the two variable models we see that `SEX` will be the most important addition on top of the SBP. And here is the summary report.

```
## How to control the summary(fit2) output to cut some junk?
## We could use packages: xtable or broom.
library(xtable)
options(xtable.comment = FALSE)
fit2 <- glm(HD~SBP + SEX, hd_data.f, family=binomial)
xtable(fit2)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-4.5703	0.3897	-11.73	0.0000
SBP	0.0187	0.0023	8.05	0.0000
SEXMALE	0.9034	0.1398	6.46	0.0000

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

```
fit2$deviance
```

```
## [1] 1373.767
```

```
fit1$deviance
```

```
## [1] 1417.468
```

```
fit2$deviance - fit1$deviance < 0
```

```
## [1] TRUE
```

Answer: Yes. Residual deviance is a goodness-of-fit statistic in a logit model. The deviance would be larger with fewer variables since we exclude the influence of other variables or meaningful interactions between, which leads to model underfitting. In `fit1`, the variable `SEX` has been restricted, meaning that β for `SEX`$= 0$, and this caused underfitting where the deviance would be larger than that without the restriction. Deviance = AIC - 2# of parameters, fit2 has smaller AIC than fit1 and fit2 has more parameter.

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?

Answer: Wald test (z-test) is shown in the summary chunk:

```r
summary(fit1)
```

```
##
## Call:
## glm(formula = HD ~ SBP, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6609  -0.7095  -0.6244  -0.5242   2.1072
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.654894   0.347875 -10.506  < 2e-16 ***
## SBP          0.015814   0.002222   7.118  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: 1421.5
##
## Number of Fisher Scoring iterations: 4
```

```r
confint(fit1, level = .99)
```

```
## Waiting for profiling to be done...

##                   0.5 %      99.5 %
## (Intercept) -4.56760794 -2.77167235
## SBP          0.01014666  0.02162211
```

```r
summary(fit2)
```

```
##
## Call:
## glm(formula = HD ~ SBP + SEX, family = binomial, data = hd_data.f)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6408  -0.7373  -0.5726  -0.4169   2.2452
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.570256   0.389727 -11.727  < 2e-16 ***
## SBP          0.018717   0.002324   8.053 8.07e-16 ***
## SEXMALE      0.903420   0.139762   6.464 1.02e-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: 1379.8
##
## Number of Fisher Scoring iterations: 4
```

```r
confint(fit2, level = .99)
```

```
## Waiting for profiling to be done...
```

```
##                   0.5 %      99.5 %
## (Intercept) -5.59732078 -3.58576526
## SBP          0.01280634  0.02480946
## SEXMALE      0.54725779  1.26824987
```

The likelihood ratio test is shown as following:

$$
\begin{aligned}
\text{Testing stat} = \chi^2 &= -2 \times \log \frac{\max_{H_1} \mathcal{L}\rangle\|(\beta_0, \beta_1|D)}{\max_{H_0} \mathcal{L}\rangle\|(\beta_0, \beta_1|D)} \\
&= -2\log(\mathcal{L}\rangle\|_{H_0}) - (-2\log(\mathcal{L}\rangle\|_{H_1})) \\
&= NullDeviance - ResidualDeviance \\
&\sim \chi^2_{df=1}
\end{aligned}
$$

The numbers are also output by the summary function:

- Null Deviance = 1469.3
- Residual Deviance = 1373.8
- $\chi^2 = 1469.3 - 1373.8 = 95.5$

```r
anova(fit1, 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     1469.3
## SBP   1   51.864      1391     1417.5 5.949e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
chi.sq <- 1469.3-1373.8
pchisq(chi.sq, 1, lower.tail = FALSE)
```

```
## [1] 1.478913e-22
```

```
anova(fit2, test="Chisq", alpha = .99)
```

```
## 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     1469.3
## SBP   1   51.864      1391     1417.5 5.949e-13 ***
## SEX   1   43.700      1390     1373.8 3.828e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:The p-value for both tests are 0.00. They are similar but not exactly the same. However, both tests demonstrate that the added variable `SEX` is significant at .01 level.

Part 1C - 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.

```
fit.full <- glm(HD~., hd_data.f, family=binomial)
summary(fit.full)
fit.full.1 <- update(fit.full, .~. -DBP)
summary(fit.full.1)
fit.full.2 <- update(fit.full.1, .~. -FRW)
summary(fit.full.2)
fit.full.3 <- update(fit.full.2, .~. -CIG)
summary(fit.full.3)

fit.full.3.predict <- predict(fit.full.3, hd_data.f, type="response")

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

Our model is:

$Logit = -8.41 + 0.056 * Age + 0.99 * SEX(Male) + 0.017 * SBP + 0.004 * CHOL$

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?

```
library(bestglm)


# Get the design matrix without 1's and HD.
fit.full <- glm(HD~., hd_data.f, family=binomial)
Xy <- model.matrix(HD ~.+0, hd_data.f)

#Attach y as the last column.
Xy <- data.frame(Xy, hd_data.f$HD)

fit.all <- bestglm(Xy, family = binomial, method = "exhaustive", IC="AIC", nvmax = 10)
```

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

```
summary(fit.all$BestModel)
```

```
##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7066  -0.7279  -0.5517  -0.3343   2.4501
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.227856   0.996153  -9.263  < 2e-16 ***
## AGE          0.061529   0.014775   4.164 3.12e-05 ***
## SEXMALE      0.911274   0.157117   5.800 6.63e-09 ***
## SBP          0.015966   0.002487   6.420 1.37e-10 ***
## CHOL         0.004493   0.001503   2.990  0.00279 **
## FRW          0.006039   0.004004   1.508  0.13151
## CIG          0.012279   0.006088   2.017  0.04369 *
## ---
## 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: 1357.3
##
## Number of Fisher Scoring iterations: 4
```

```
fit.final = fit.all$BestModel
```

Answer: No. We observed that in our best model, `FRW` is not significant at .05 level since the p-value is larger than .05. Obviously, our final model is not the same as the model from backwards elimination.

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".

Answer: Our model is:

$Logit = -9.23 + 0.06 * AGE + 0.91 * SEX(Male) + 0.016 * SBP + 0.004 * CHOL + 0.006 * FRW + 0.012 * CIG$

From the model we can say that, collectively, AGE, SBP, CHOL, CIG are all positively related to the chance of a HD, although the correlations are weak (the slopes are mild). Specifically, the log of having a heart disease would incrase by 0.016 if SBP increased by 1. The other variables can be interperated in the same way. Also, Males would have a higher chance of having heart disease than Females while all the other factors are controlled in our model.

Notice: Here, we exclude FRW as an important predictor in our model because the p-value of z score is not significant at a .05 level.

Part 1D - Prediction

Liz is a patient with the following readings: AGE=50, GENDER=FEMALE, SBP=110, DBP=80, CHOL=180, FRW=105, CIG=0. What is the probability that she will have heart disease, according to our final model?

```
no <- 1/(1+exp(-9.23+0.06*50+0.016*110+0.004*180+0.006*105+0.012*0))
yes <- 1-no
yes
```

```
## [1] 0.04228977
```

Answer: The probability that Liz will have heart diease is 4.2% according to our final model.

Part 2 - Classification analysis

a. 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.

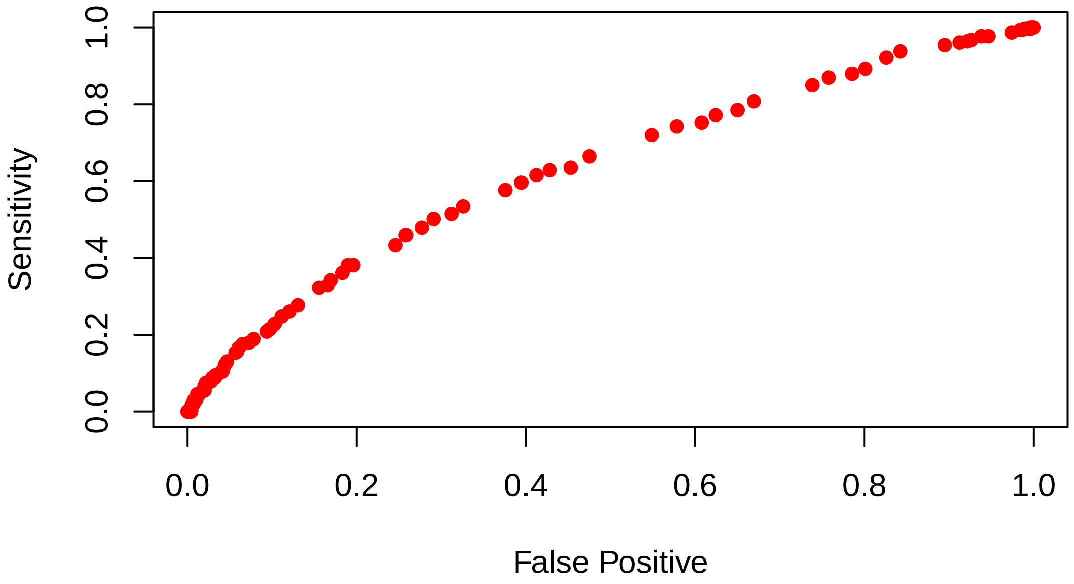
```
library(pROC)
```

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

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

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

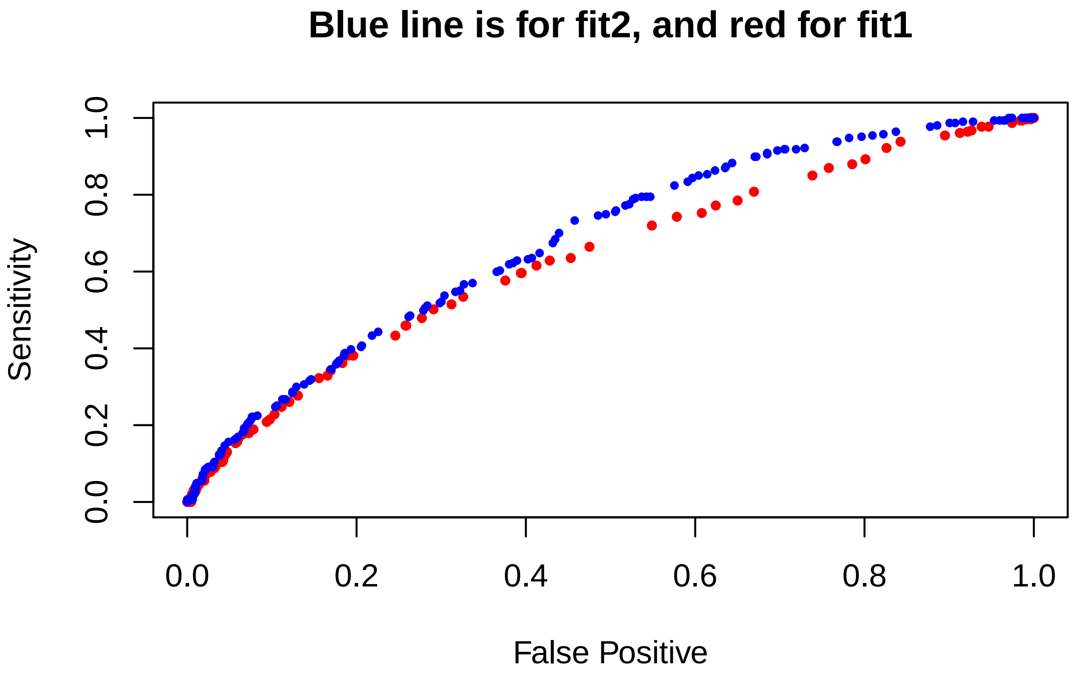
```
fit1.roc<- roc(hd_data.f$HD, fit1$fitted, col="blue")

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



Answer: ROC curvers measure the performance of such a classifier. ROC curve is the True positive$P(\hat{Y} = 1|Y = 1)$ against False Positive$P(\hat{Y} = 1|Y = 0)$, the higher true positive will lead to higher False Positive, if true positive is 0, then false positive is also 0. In contrast, if true positive is 1, then false positive is also 1. We can also use AUC (Area under the curve) to measure. It is also used to measure the performance of the classifier as a whole: the larger the better.

b. 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?

```
fit2.roc<- roc(hd_data.f$HD, fit2$fitted, col="blue")
plot(1-fit1.roc$specificities, fit1.roc$sensitivities, col="red", pch=16, cex=.7,
     xlab="False Positive",
     ylab="Sensitivity")
points(1-fit2.roc$specificities, fit2.roc$sensitivities, col="blue", pch=16, cex=.6)
title("Blue line is for fit2, and red for fit1")
```



Answer:

Yes,fit2 is always contain fit1 and AUC of fit2 is always larger than AUC of fit1. Because ROC increases with more variable, since ROC measures the training data, so it may occur overfitting.

c. 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?

```
#Positive Prediction for fit 1
fit1.pred.5 <- ifelse(fit1$fitted.values > 0.5, "1", "0")
cm1.5 <- table(fit1.pred.5, hd_data.f$HD)
cm1.5
```

```
##
## fit1.pred.5    0    1
##           0 1075  298
##           1   11    9
```

```
positive1.pred <- cm1.5[2, 2] / (cm1.5[2, 1] + cm1.5[2, 2])
positive1.pred
```

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

```
#Negative Prediction for fit2
negative1.pred <- cm1.5[1, 1] / (cm1.5[1, 1] + cm1.5[1, 2])
negative1.pred
```

```
## [1] 0.782957
```

```
#Positive Prediction for fit 2
fit2.pred.5 <- ifelse(fit2$fitted.values > 0.5, "1", "0")
cm2.5 <- table(fit2.pred.5, hd_data.f$HD)
cm2.5
```

```
##
## fit2.pred.5    0    1
##           0 1067  290
##           1   19   17
```

```
positive2.pred <- cm2.5[2, 2] / (cm2.5[2, 1] + cm2.5[2, 2])
positive2.pred
```

```
## [1] 0.4722222
```

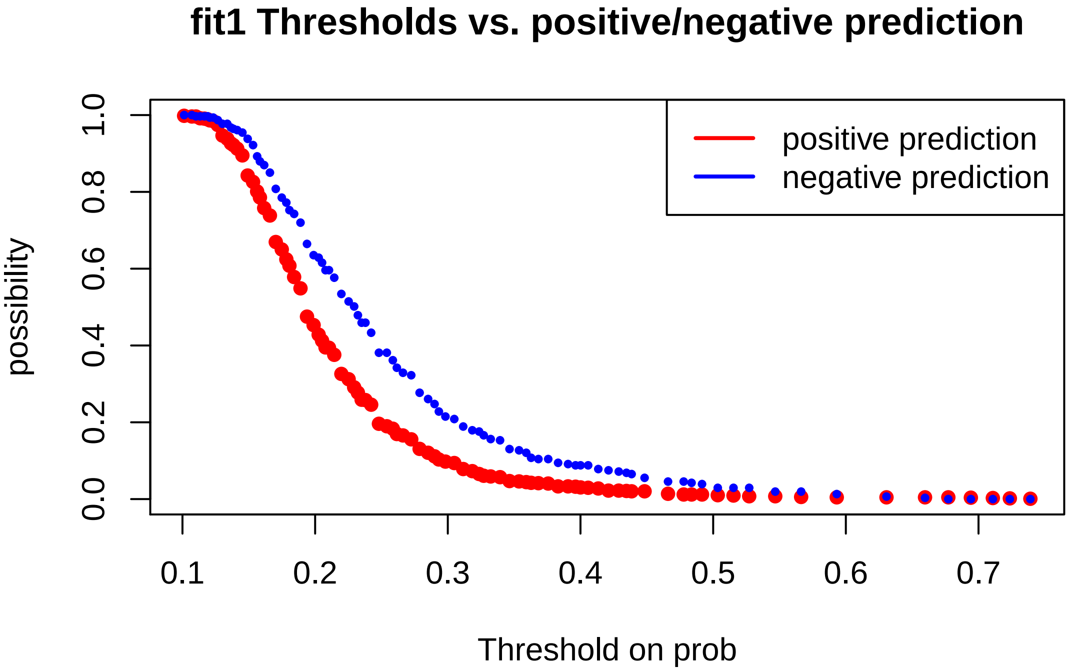
```
#Negative Prediction
negative2.pred <- cm2.5[1, 1] / (cm2.5[1, 1] + cm2.5[1, 2])
negative2.pred
```

```
## [1] 0.7862933
```

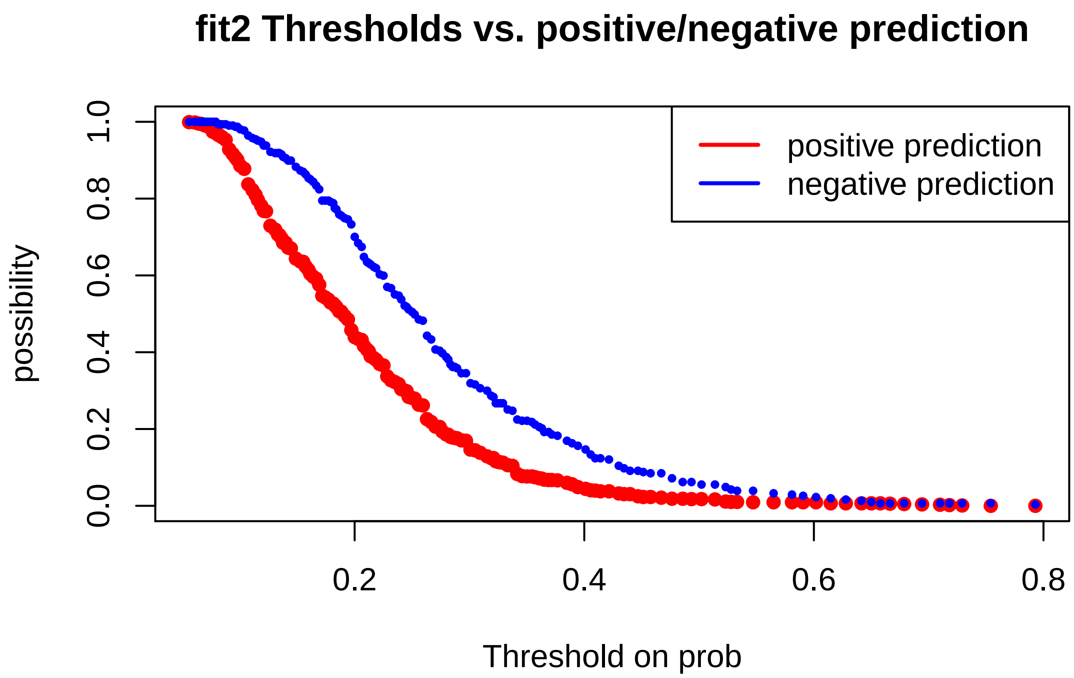
Answer: fit2 is more desirable if we prioritize the Positive Prediction values. d. (Optional/extra credit) For `fit1`: overlay two curves, but put the threshold over the probability function as the x-axis and positive prediction values and the negative prediction values as the y-axis. Overlay the same 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(pROC)
fit1.roc = roc(hd_data.f$HD, fit1$fitted, col="blue")
plot(fit1.roc$thresholds, 1-fit1.roc$specificities,  col="red", pch=16,
     xlab="Threshold on prob",
     ylab="possibility",
     main = "fit1 Thresholds vs. positive/negative prediction")
```

```
points(fit1.roc$thresholds, fit1.roc$sensitivities, col="blue", pch=16, cex=.6)
legend("topright", legend=c("positive prediction", "negative prediction"),
       lty=c(1,1), lwd=c(2,2), col=c("red", "blue"))
```



```
fit2.roc = roc(hd_data.f$HD, fit2$fitted, col="blue")
plot(fit2.roc$thresholds, 1-fit2.roc$specificities,  col="red", pch=16,
     xlab="Threshold on prob",
     ylab="possibility",
     main = "fit2 Thresholds vs. positive/negative prediction")
points(fit2.roc$thresholds, fit2.roc$sensitivities, col="blue", pch=16, cex=.6)
legend("topright", legend=c("positive prediction", "negative prediction"),
       lty=c(1,1), lwd=c(2,2), col=c("red", "blue"))
```



### Part 3 - 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 1 B) to build a class of linear classifiers.

a. Write down the linear boundary for the Bayes classifier if the risk ratio of $a_{10}/a_{01} = 10$. **Answer**: $\frac{P(Y=1|X)}{P(Y=0|X)} > \frac{a_{01}}{a_{10}}$ $logit > log(\frac{0.09}{0.90}) = -2.30$ $logit = -9.23 + 0.06 * AGE + 0.91 * SEX(Male) + 0.016 * SBP + 0.004 * CHOL + 0.006 * FRW + 0.012 * CIG > -2.30$ Linear boundary: $0.06 * AGE + 0.91 * SEX(Male) + 0.016 * SBP + 0.004 * CHOL + 0.006 * FRW + 0.012 * CIG > 6.93$

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

```
fit.final.pred.bayes <- rep("0", length(hd_data.f$HD))
fit.final.pred.bayes[fit.final$fitted > 0.09] = "1"
fit.final.pred.bayes <- as.factor(ifelse(fit.final$fitted > 0.09, "1", "0"))
MCE.bayes=(sum(10*(fit.final.pred.bayes[hd_data.f$HD == "1"] != "1"))
           + sum(fit.final.pred.bayes[hd_data.f$HD == "0"] != "0"))/length(hd_data.f$HD)
MCE.bayes
```

```
## [1] 0.7164393
```

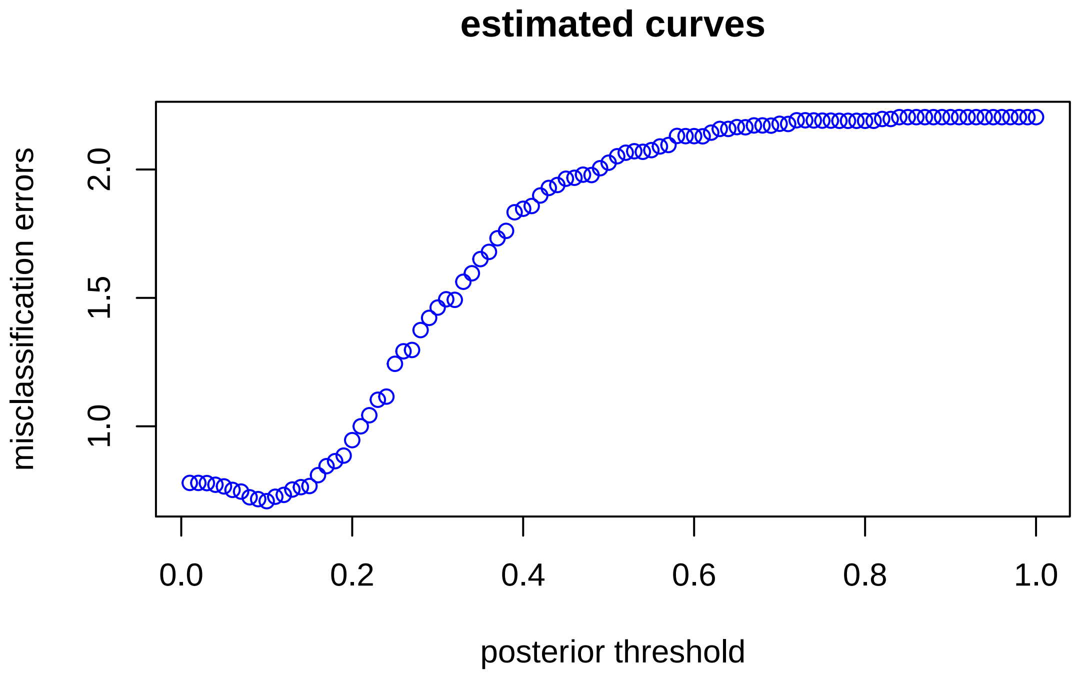
c. Recall Liz, our patient from part 1. How would you classify her under this classifier?

```
0.06*50+0.016*110+0.004*180+0.006*105+0.012*0 - 6.93 >0
```

```
## [1] FALSE
```

Answer: Liz will be classified in 0, she will not have Heart Disease. Now, draw two estimated curves where x = posterior threshold, and y = misclassification errors, corresponding to the thresholding rule given in x-axis.

```
X = rep(0,100)
Y = rep(0,100)

for (i in 1:100){
  x =i/100
  fit.pred.test <- rep("0", length(hd_data.f$HD))
  fit.pred.test[fit.final$fitted >x] = "1"
  y=(sum(10*(fit.pred.test[hd_data.f$HD == "1"] != "1"))
           + sum(fit.pred.test[hd_data.f$HD == "0"] != "0"))/length(hd_data.f$HD)
  X[i] = x
  Y[i] = y
}
plot(X,Y,col = "blue",xlab = "posterior threshold",ylab = "misclassification errors")
title("estimated curves")
```



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

```
fit.final.pred.bayes.10 <- rep("0", length(hd_data.f$HD))
fit.final.pred.bayes.10[fit.final$fitted > 0.09] = "1"

MCE.bayes.10=(sum(10*(fit.final.pred.bayes.10[hd_data.f$HD == "1"] != "1"))
           + sum(fit.final.pred.bayes.10[hd_data.f$HD == "0"] != "0"))/length(hd_data.f$HD)
MCE.bayes.10
```

```
## [1] 0.7164393
```

```
#
```

Answer: The MCE is small, so the Bayes rule classifier performs well.

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

```
fit.final.pred.bayes2 <- rep("0", length(hd_data.f$HD))
fit.final.pred.bayes2[fit.final$fitted > 0.5] = "1"
MCE_1.bayes=(sum(1*(fit.final.pred.bayes2[hd_data.f$HD == "1"] != "1"))
           + sum(fit.final.pred.bayes2[hd_data.f$HD == "0"] != "0"))/length(hd_data.f$HD)
MCE_1.bayes
```

```
## [1] 0.2175162
```

Answer: The Misclassification error is very low, so the Bayes rule classifier performs well.

Problem 2

How well can we predict whether a bill will be passed by the legislature?

Hundreds to thousands of bills are written each year in Pennsylvania. Some are long, others are short. Most of the bills do not even get to be voted on ("sent to the floor"). The chamber meets for 2-year sessions. Bills that are not voted on before the end of the session (or which are voted on but lose the vote) are declared dead. Most bills die. In this study we examine about 8000 bills proposed since 2009, with the goal of building a classifier which has decent power to forecast which bills are likely to be passed.

We have available some information about 8011 bills pertaining to legislation introduced into the Pennsylvania House of Representatives. The goal is to predict which proposals will pass the House. Here is some information about the data:

The response is the variable called `status`. `Bill:passed` means that the bill passed the House; `governor:signed` means that the bill passed both chambers (including the House) and was enacted into law; `governor:received` means that the bill has passed both chambers and was placed before the governor for consideration. All three of these statuses signify a success or a PASS (Meaning that the legislature passed the bill. This does not require it becoming law). All other outcomes are failures.

Here are the rest of the columns:

- `Session` – in which legislative session was the bill introduced
- `Sponsor_party` – the party of the legislator who sponsored the bill (every bill has a sponsor)
- `Bill_id` – of the form HB-[bill number]-[session], e.g., `HB-2661-2013-2014` for the 2661st House Bill introduced in the 2013-2014 session.
- `Num_cosponsors` – how many legislators cosponsored the bill
- `Num_d_cosponsors` – how many Democrats cosponsored the bill
- `Num_r_cosponsors` – how many Republicans cosponsored the bill
- `Title_word_count` – how many words are in the bill's title
- `Originating_committee` – most bills are sent ("referred") to a committee of jurisdiction (like the transportation committee, banking & insurance committee, agriculture & rural affairs committee) where they are discussed and amended. The originating committee is the committee to which a bill is referred.
- `Day_of_week_introduced` – on what day the bill was introduced in the House (1 is Monday)
- `Num_amendments` – how many amendments the bill has
- `Is_sponsor_in_leadership` – does the sponsor of the bill hold a position inside the House (such as speaker, majority leader, etc.)
- `num_originating_committee_cosponsors` – how many cosponsors sit on the committee to which the bill is referred
- `num_originating_committee_cosponsors_r` – how many Republican cosponsors sit on the committee to which the bill is referred
- `num_originating_committee_cosponsors_d` - how many Democratic cosponsors sit on the committee to which the bill is referred

The data you can use to build the classifier is called `Bills.subset`. It contains 7011 records from the full data set. I took a random sample of 1000 bills from the 2013-2014 session as testing data set in order to test the quality of your classifier, it is called `Bills.subset.test.`

Your job is to choose a best set of classifiers such that

- The testing ROC curve pushes to the upper left corner the most, and has a competitive AUC value.
- Propose a reasonable loss function, and report the Bayes rule together with its weighted MIC.
- You may also create some sensible variables based on the predictors or make other transformations to improve the performance of your classifier.

Here is what you need to report:

1. Write a summary about the goal of the project. Give some background information. If desired, you may go online to find out more information.

2. Give a preliminary summary of the data.
3. Based on the data available to you, you need to build a classifier. Provide the following information:
    - The process of building your classifier
    - Methods explored, and why you chose your final model
    - Did you use a training and test set to build your classifier using the training data? If so, describe the process including information about the size of your training and test sets.
    - What is the criterion being used to build your classifier?
    - How do you estimate the quality of your classifier?
4. Suggestions you may have: what important features should have been collected which would have helped us to improve the quality of the classifiers.

Final notes: The data is graciously lent from a friend. It is only meant for you to use in this class. All other uses are prohibited without permission.

Answer: See report with R code starting the next page.

Project Goal

In an era of government, Congress is where people write thousands of bills to each year but criticized to be unproductive. As mentioned in the news of *TheWashingtonPost* on Feb 1, 2018, members introduce about 11,000 pieces of legislation in a typical two-year term. A few hundred come to a floor vote, and only about half of those will be signed into law. Therefore, it brings attention to people what factors are involved in the success of bills. To investigate this, we develop a model to predict whether a bill will be successfully passed by the legislature using Pennsylvania as a representative. We're going to show which factors are worth paying attention to by giving them a priority for people to consider before they submit the bills.

Preliminary Summary of Data

```r
# read the training and testing data
train_data <- read.csv("Bills.subset.csv")
test_data <- read.csv("Bills.subset.test.csv")
```

The whole data we use in this project containing 8011 bills which pertain to legislation introduced into the Pennsylvania House of Representatives. It is splitted into training and testing data, namely `Bills.subset` and `Bills.subset.test`, respectively. The training set contains 7011 records from the full data set, and the testing data set is a random sample of 1000 bills token from the 2013-2014 session in order to test the quality of the classifier. However, some of them has missing values. At a first glance of the dataset, we find that most of the missing values come from the `Originating_committee` predictor, which indicates the committee to which a bill is referred. To find out which columns contain missing value, we assume that missing data is coded as NA or is an empty string.

```r
# find the number of missing data in training and testing data
sum(is.na(train_data))
sum(is.na(test_data))

# see which columns have missing values in training and testing set
sapply(train_data, function(x) any(is.na(x)))
sapply(train_data, function(x) any(x == ""))

sapply(train_data, function(x) any(is.na(x)))
sapply(train_data, function(x) any(x == ""))
```

We found that missing values are included in columns `day.of.week.introduced`, `status`, `sponsor_party`, and `originating_committee`. To git rid of all these columns, we first assign all empty cells in both training and testing set to NA, and then omit all the missing data by applying `na.omit`.

```r
train_data[train_data==""] <- NA
test_data[test_data==""] <- NA

train_data <- na.omit(train_data)
test_data <- na.omit(test_data)
```

The final training set contains 6647 instances out of the original 7011 instances, and the final testing set contains 999 instead of the original 1000 instances.

The response is the variable called `status.` It includes Nine statuses which are summarized as follow.

Bill Status	Success	Description
`bill:passed`	Yes	Passed the House
`governor:signed`	Yes	Passed both chambers (including the House) and was enacted into law

Bill Status	Success	Description
governor:received	Yes	Passed both chambers and was placed before the governor for consideration
committee:referred	No	The bill is referred to the commitee
committee:passed	No	The bill passed the commitee
amendment:passed	No	The bill passed the amendment
bill:reading:1	No	The bill is being reading
bill:reading:2	No	The bill is being reading
bill:reading:3	No	The bill is being reading

We signify statuses bill:passed, governor:signed, and governor:received as a success or a PASS (Meaning that the legislature passed the bill. This does not require it becoming law) while all other outcomes as failures. The final training set contains 455 of successful bills and 6192 of failures, and the final testing set includes 68 successes and 931 failures.

```r
# rename statuses `withbill:passed`, `governor:signed`, and `governor:received` as 1, otherwise 0
levels(train_data$status)[levels(train_data$status) == "bill:passed"] <- 1
levels(train_data$status)[levels(train_data$status) == "governor:signed"] <- 1
levels(train_data$status)[levels(train_data$status) == "governor:received"] <- 1
levels(train_data$status)[levels(train_data$status) != 1] <- 0

levels(test_data$status)[levels(test_data$status) == "bill:passed"] <- 1
levels(test_data$status)[levels(test_data$status) == "governor:signed"] <- 1
levels(test_data$status)[levels(test_data$status) == "governor:received"] <- 1
levels(test_data$status)[levels(test_data$status) != 1] <- 0

# count the number of instances of successes and failures
sum(train_data$status == 1) #455
sum(train_data$status == 0)  #6192

sum(test_data$status == 1) #68
sum(test_data$status == 0)  #931
```

There are 14 predictors that are taken into considering in the original dataset described in the table below, with some of them continuous and others categorical.

Predictors	Description
Session	In which legislative session was the bill introduced
Sponsor_party	The party of the legislator who sponsored the bill (every bill has a sponsor)
Bill_id	Of the form HB-[bill number]-[session], e.g., HB-2661-2013-2014 for the 2661st House Bill introduced in the 2013-2014 session.
Num_cosponsors	How many legislators cosponsored the bill
Num_d_cosponsors	How many Democrats cosponsored the bill
Num_r_cosponsors	How many Republicans cosponsored the bill
Title_word_count	How many words are in the bill's title
Originating_committee	Most bills are sent ("referred") to a committee of jurisdiction (like the transportation committee, banking & insurance committee, agriculture & rural affairs committee) where they are discussed and amended. The originating committee is the committee to which a bill is referred.
Day_of_week_introduced	On what day the bill was introduced in the House (1 is Monday)
Num_amendments	How many amendments the bill has
Is_sponsor_in_leadership	Does the sponsor of the bill hold a position inside the House (such as speaker, majority leader, etc.)

Predictors	Description
`num_originating_committee_cosponsors`	How many cosponsors sit on the committee to which the bill is referred
`num_originating_committee_cosponsors_r`	How many Republican cosponsors sit on the committee to which the bill is referred
`num_originating_committee_cosponsors_d`	How many Democratic cosponsors sit on the committee to which the bill is referred

Classification and Model Selection

First, we use logistic regression on our train dataset. By looking at the structure of the predictors, we find that `bill_id` is a factor of 7011 levels, which means that each instances has different values. We also find that the the years in `bill_id` match the predictor `session`. Therefore, we can git rid of the predictor `bill_id` since it does not give much useful information to our model.

```
str(train_data)
```

```
## 'data.frame':    6647 obs. of  15 variables:
##  $ bill_id                               : Factor w/ 7011 levels "HB-1-2009-2010",..: 3720 4699 3055
##  $ sponsor_party                         : Factor w/ 3 levels "","Democratic",..: 2 2 2 2 2 2 2 3 3
##  $ session                               : Factor w/ 4 levels "2009-2010","2009-2010 Special Session
##  $ num_cosponsors                        : int  0 9 30 4 0 4 30 19 47 15 ...
##  $ num_d_cosponsors                      : int  0 6 24 4 0 3 20 2 13 14 ...
##  $ num_r_cosponsors                      : int  0 3 6 0 0 1 10 17 34 1 ...
##  $ title_word_count                      : int  50 20 25 29 30 25 28 48 23 29 ...
##  $ originating_committee                 : Factor w/ 26 levels "","PAC000001",..: 6 3 11 3 8 3 2 16
##  $ day.of.week.introduced                : int  5 1 1 1 3 2 2 1 1 2 ...
##  $ num_amendments                        : int  0 0 1 0 0 0 1 0 0 1 ...
##  $ status                                : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 2 ...
##  $ is_sponsor_in_leadership              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ num_originating_committee_cosponsors  : int  0 3 3 0 0 0 1 1 2 2 ...
##  $ num_originating_committee_cosponsors_r: int  0 1 1 0 0 0 1 1 2 0 ...
##  $ num_originating_committee_cosponsors_d: int  0 2 2 0 0 0 0 0 0 2 ...
##  - attr(*, "na.action")= 'omit' Named int  1 7 9 10 19 28 32 33 38 41 ...
##   ..- attr(*, "names")= chr  "1" "7" "9" "10" ...
```

We also find that `num_cosponsors` is correlated to `num_d_cosponsors` and `num_r_cosponsors` (`num_cosponsors` = `num_d_cosponsors` + `num_r_cosponsors`), so we can get the result of one of them given the other two values. So do the predictors `num_originating_committee_cosponsors`, `num_originating_committee_cosponsors_d`, and `num_originating_committee_cosponsors_r`. Also, we find that only the ingredient of the committee matters, not the title, so we drop `originating_committee`. Thus the first model includes all predictors except `bill_id`, `num_r_cosponsors`, `num_originating_committee_cosponsors_d`, and `originating_committee`.

We use `Anova()` to drop categorical predictor which has low p-value. Then use backward selection method to keep only variables whose coefficients are significantly different from 0 at .05 level, and kick out the variable with the largest p-value first, and then re-fit the model to see if there are other variables should be kicked out.

```
library(car)
```

```
## Loading required package: carData
```

```
fit.bill <- glm(status~.-bill_id -num_r_cosponsors -num_originating_committee_cosponsors_d
                -originating_committee, train_data, family=binomial)
summary(fit.bill)
```

```r
Anova(fit.bill)

fit.bill.1 <- update(fit.bill, .~. -num_cosponsors)
summary(fit.bill.1)
fit.bill.2 <- update(fit.bill.1, .~. -num_originating_committee_cosponsors_r)
summary(fit.bill.2)
fit.bill.3 <- update(fit.bill.2, .~. -num_d_cosponsors)
summary(fit.bill.3)
fit.bill.4 <- update(fit.bill.3, .~. -day.of.week.introduced)
summary(fit.bill.4)
fit.bill.5 <- update(fit.bill.4, .~. -is_sponsor_in_leadership)
summary(fit.bill.5)
fit.bill.6 <- update(fit.bill.5, .~. -num_originating_committee_cosponsors)
summary(fit.bill.6)
```

```r
fit.bill.6.predict <- predict(fit.bill.6, train_data, type="response")
fit.bill.backward <- glm(status~sponsor_party+session+title_word_count+num_amendments,
                         train_data, family = binomial)
summary(fit.bill.backward)
```

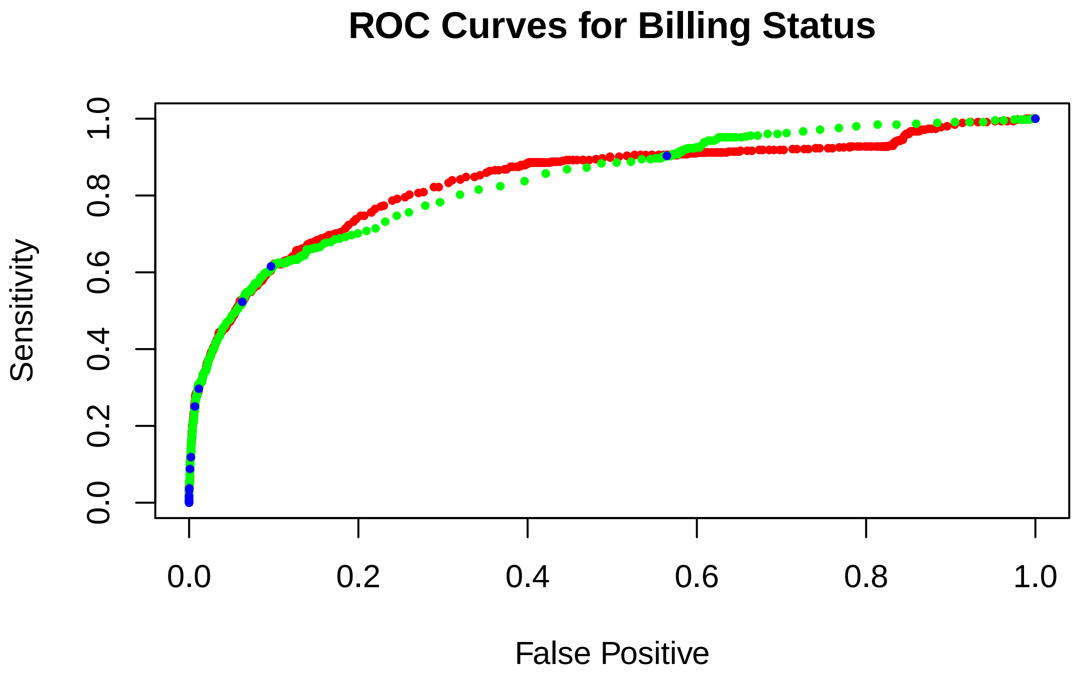
```
##
## Call:
## glm(formula = status ~ sponsor_party + session + title_word_count +
##     num_amendments, family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7280  -0.3109  -0.2502  -0.2135   2.9013
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -4.269217   0.137475 -31.054  < 2e-16 ***
## sponsor_partyRepublican  0.726797   0.129537   5.611 2.01e-08 ***
## session2011-2012         0.419209   0.140381   2.986  0.00282 **
## session2013-2014         0.415172   0.158068   2.627  0.00863 **
## title_word_count         0.004709   0.001097   4.292 1.77e-05 ***
## num_amendments           1.785032   0.077224  23.115  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3318.4  on 6646  degrees of freedom
## Residual deviance: 2456.3  on 6641  degrees of freedom
## AIC: 2468.3
##
## Number of Fisher Scoring iterations: 6
```

From the model we can say that, collectively, `title_word_count`, `num_amendments` are all positively related to the chance of a `status`, although the correlations are weak (the slopes are mild). Specifically, the log of status would increase by 1.7850 if `num_amendments` increased by 1. The other variables can be interperated in the same way. Also, `Republican` would have a higher chance of success than `Democratic` while all the other factors are controlled in our model, and the success also varies each session year.

Prediction and Analysis

We do prediction on the testing data using the model we obtain from training data. And display ROC curve and AUC (Area under the curve) to select the best classifier.

```
fit.bill.predict.train <- predict(fit.bill.backward, train_data, type="response")
fit.bill.predict.test <- predict(fit.bill.backward, test_data, type="response")

library(pROC)
fit.bill.final.1 <- glm(status~sponsor_party+session+title_word_count+num_amendments,
                        train_data, family = binomial)
fit.bill.final.2 <- glm(status~sponsor_party+title_word_count+num_amendments, train_data,
                        family = binomial)
fit.bill.final.3 <- glm(status~sponsor_party+num_amendments, train_data, family = binomial)

bill.fit1.roc<- roc(train_data$status, fit.bill.final.1$fitted, col="green")
bill.fit2.roc<- roc(train_data$status, fit.bill.final.2$fitted, col="red")
bill.fit3.roc<- roc(train_data$status, fit.bill.final.3$fitted, col="blue")


plot(1-bill.fit1.roc$specificities, bill.fit1.roc$sensitivities, col="red", pch=16, cex=.6,
     xlab="False Positive",
     ylab="Sensitivity")
points(1-bill.fit2.roc$specificities, bill.fit2.roc$sensitivities, col="green", pch=16, cex=.6)
points(1-bill.fit3.roc$specificities, bill.fit3.roc$sensitivities, col="blue", pch=16, cex=.6)

title("ROC Curves for Billing Status")
```



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

```
## Area under the curve: 0.8363
```

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

```
## Area under the curve: 0.8353
```

```
pROC::auc(bill.fit3.roc)
```

```
## Area under the curve: 0.8123
```

We create the confusion matrix table to estimate the Positive Prediction Values and Negative Prediction Values using .5 as a threshold.

```
#Positive Prediction
bill.fit.pred <- ifelse(fit.bill.final.1$fitted.values > 0.5, "1", "0")
bill.cm <- table(bill.fit.pred, train_data$status)
bill.cm
```

```
##
## bill.fit.pred    0    1
##             0 6147  334
##             1   45  121
```

```
positive.pred <- bill.cm[2, 2] / (bill.cm[2, 1] + bill.cm[2, 2])
positive.pred
```

```
## [1] 0.7289157
```

```
#Negative Prediction
negative.pred <- bill.cm[1, 1] / (bill.cm[1, 1] + bill.cm[1, 2])
negative.pred
```

```
## [1] 0.9484647
```

We use Bayes rules with risk ratio $\frac{a_{10}}{a_{01}} = 5$ or $\frac{a_{10}}{a_{01}} = 1$, and use our final model to build a class of linear classifiers. The MCE is small, so the Bayes rule classifier performs well.

```
bill.fit.final.pred.bayes <- rep("0", length(train_data$status))
bill.fit.final.pred.bayes[fit.bill.final.1$fitted > 0.09] = "1"
bill.fit.final.pred.bayes <- as.factor(ifelse(fit.bill.final.1$fitted > 0.09, "1", "0"))
MCE.bill.bayes=(sum(5*(bill.fit.final.pred.bayes[train_data$status == "1"] != "1"))
           + sum(bill.fit.final.pred.bayes[train_data$status == "0"] != "0"))/length(train_data$status)
MCE.bill.bayes
```

```
## [1] 0.2225064
```

```
bill.fit.final.pred.bayes2 <- rep("0", length(train_data$status))
bill.fit.final.pred.bayes2[fit.bill.final.1$fitted > 0.09] = "1"
bill.fit.final.pred.bayes2 <- as.factor(ifelse(fit.bill.final.1$fitted > 0.09, "1", "0"))
MCE.bill.bayes2=(sum(1*(bill.fit.final.pred.bayes2[train_data$status == "1"] != "1"))
           + sum(bill.fit.final.pred.bayes2[train_data$status == "0"] != "0"))/length(train_data$status
MCE.bill.bayes2
```

```
## [1] 0.105762
```