PROJECT 6 STAT 5474

Prince Appiah

11/18/2021

Read in data

```
ILPD <- read.csv(file = "Indian Liver Patient Dataset (ILPD).csv",
       header=FALSE, col.names=c("age", "gender", "TB", "DB", "alkphos",
       "sgpt", "sgot", "TP", "alb", "AGratio", "liver"))
    dim(ILPD); head(ILPD)
```

```
## [1] 583  11
```

```
##   age gender   TB  DB alkphos sgpt sgot  TP alb AGratio liver
## 1  65 Female  0.7 0.1     187   16   18 6.8 3.3    0.90     1
## 2  62   Male 10.9 5.5     699   64  100 7.5 3.2    0.74     1
## 3  62   Male  7.3 4.1     490   60   68 7.0 3.3    0.89     1
## 4  58   Male  1.0 0.4     182   14   20 6.8 3.4    1.00     1
## 5  72   Male  3.9 2.0     195   27   59 7.3 2.4    0.40     1
## 6  46   Male  1.8 0.7     208   19   14 7.6 4.4    1.30     1
```

Data Cleaning

(a) What is the proportion of subjects who were diagnosed with liver diseases? Do you think this is ever close to the real prevalence rate of liver diseases in the general population?

```
prop <- data.frame(table(ILPD$liver))
prop$Proportion <- (prop$Freq/sum(prop$Freq))*100
names(prop) <- c("liver","Frequency","Proportion")
prop
```

```
##   liver Frequency Proportion
## 1     1       416   71.35506
## 2     2       167   28.64494
```

Comment We see from the table that a higher percentage(71.36%) of the subjects have liver disease. This rate may be too far from the real prevalence rate of the liver diseases in the general population.

(b) Are there any missing data? If so, handle them in an appropriate way (via, e.g., listwise deletion, imputation)

```r
# Checking for missing data
library(questionr)
freq.na(ILPD)
```

```
##         missing %
## AGratio       4 1
## age           0 0
## gender        0 0
## TB            0 0
## DB            0 0
## alkphos       0 0
## sgpt          0 0
## sgot          0 0
## TP            0 0
## alb           0 0
## liver         0 0
```

Comment We see from the output that the variable has 4 missing values with approximately 1% of the data.

```r
# Imputation by Mice
set.seed(123)
suppressPackageStartupMessages(library(mice))
data.imputed <- mice(ILPD, printFlag = F)
```

```
## Warning: Number of logged events: 1
```

```r
dat <- complete(data.imputed, 1)
dat <- as.data.frame(dat)

# change the value 2 of liver to 0
cond <- dat$liver == 1
dat$liver[!cond] <- 0

# Verifying that there are no missing values after the imputation
freq.na(dat)
```

```
##         missing %
## age           0 0
## gender        0 0
## TB            0 0
## DB            0 0
## alkphos       0 0
## sgpt          0 0
## sgot          0 0
## TP            0 0
## alb           0 0
## AGratio       0 0
## liver         0 0
```

Comment we see that after the imputation by mice there are no missing values in the data.

EDA and Variable Screening

(a) Among all the predictors, how many of are continuous, integer counts and categorical?

```
str(dat)
```

```
## 'data.frame':    583 obs. of  11 variables:
##  $ age    : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ gender : chr  "Female" "Male" "Male" "Male" ...
##  $ TB     : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ DB     : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ alkphos: int  187 699 490 182 195 208 154 202 202 290 ...
##  $ sgpt   : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ sgot   : int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP     : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ alb    : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ AGratio: num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ liver  : num  1 1 1 1 1 1 1 1 0 1 ...
```

Comment From the output we have 5 continuous counts, 4 integer counts and 2 categorical counts.

(b) For each categorical predictor, use X2 -test of independence to assess its association with the binary response liver. For other types of predictors, use either two-sample t test (or its nonparametric alternative – Wilcoxon rank-sum test). Output the p-value for each variable. Some sample R code for this part is given below. Alternatively, you may use simple logistic regression for this purpose.

subsection{Test of Independent and Two Sample t-test}

```
# Two sample t-test
cond.1 <- dat$liver == 1
cond.2 <- as.vector(which(sapply(dat[,-c(11)], is.numeric), arr.ind = T))
print("Test of Normality of the Numerical Variables for Subjects Diagnosed as Liver Disease")
```

```
## [1] "Test of Normality of the Numerical Variables for Subjects Diagnosed as Liver Disease"
```

```
sapply(dat[cond.1, cond.2], shapiro.test)[-c(3,4),]
```

```
##           age        TB           DB           alkphos      sgpt
## statistic 0.9901649  0.5281952    0.6060369    0.6223847    0.373496
## p.value   0.00698313 7.622218e-32 1.094357e-29 3.424658e-29 2.129166e-35
##           sgot        TP         alb        AGratio
## statistic 0.3191012   0.9910701  0.9927984  0.9288662
## p.value   1.76763e-36 0.01294119 0.04315425 3.623808e-13
```

```
print("Test of Normality of the Numerical Variables for Subjects without Liver Disease")
```

```
## [1] "Test of Normality of the Numerical Variables for Subjects without Liver Disease"
```

```r
sapply(dat[!cond.1, cond.2], shapiro.test)[-c(3,4),]
```

```
##           age        TB           DB           alkphos      sgpt
## statistic 0.983243   0.5001996    0.5073643    0.4961543    0.685253
## p.value   0.04158211 1.356559e-21 1.858271e-21 1.137452e-21 1.898717e-17
##           sgot         TP        alb        AGratio
## statistic 0.6364154    0.9894444 0.9818292  0.9682523
## p.value   1.087504e-18 0.2485639 0.02760179 0.0007129075
```

Comment For the numerical variables, we first use Shapiro-Wilk test to check the assumption of normality so as to know whether to use parametric or nonparametric approach for the two sample t-test. We see from the output of the Shapiro-Wilk normality test above that the assumption of normality is violated since the p-values are less than 0.05 in each group with the exception of the variable ‘TP’ for subjects without liver disease. Thus, we use the Wilcoxon rank-sum test.

```r
set.seed(123)
suppressPackageStartupMessages(library(car))
vars.nominal <- c("gender")
cols.x <- 1:(NCOL(dat)-1)
xnames <- names(dat)[cols.x]
y <- dat$liver
OUT <- NULL
for (j in 1:length(cols.x)){
  x <- dat[, cols.x[j]]
  xname <- xnames[j]
  if (is.element(xname, vars.nominal)){
    tbl <- table(x, y)
    pvalue <- chisq.test(tbl)$p.value
  } else {
    # WILCOXON TEST
    pvalue <- wilcox.test(x~y, alternative="two.sided")$p.value
  }
  OUT <- rbind(OUT, cbind(xname=xname, pvalue=pvalue))
}
OUT <- as.data.frame(OUT, stringsAsFactors =F)
colnames(OUT) <- c("name", "pvalue")
OUT
```

```
##       name               pvalue
## 1      age  0.00177436824852034
## 2   gender   0.0596658468577747
## 3       TB 2.28951895520505e-13
## 4       DB 7.43112588798556e-13
## 5  alkphos  4.3472343156702e-11
## 6     sgpt 2.33293513824168e-12
## 7     sgot 9.20966244908938e-14
## 8       TP    0.437146647592439
## 9      alb 5.56700396782932e-05
## 10 AGratio 9.01436082316323e-06
```

(c) Applying a liberal threshold significance level a = 0.20, exclude predictors that are associated with a p-value larger than that from the subsequent logistic model fitting.

```
#Variable screening
cond.3 <- OUT$pvalue > 0.20
OUT[cond.3, ]
```

```
##       name               pvalue
## 3       TB 2.28951895520505e-13
## 4       DB 7.43112588798556e-13
## 5  alkphos  4.3472343156702e-11
## 6     sgpt 2.33293513824168e-12
## 7     sgot 9.20966244908938e-14
## 8       TP    0.437146647592439
## 9      alb 5.56700396782932e-05
## 10 AGratio 9.01436082316323e-06
```

Comment From the output,the predictor TP has a p-value greater than 0.20. Therefore, we exclude TP from the subsequent analysis.

Variable Selection

(a) First fit the full model with all predictors that have passed the screening in Part 2c. Call it fit.full.

```
#Full model.
set.seed(123)
formula0 <- liver ~ age + factor(gender) + TB + DB + alkphos + sgpt +
          sgot + alb  + AGratio
fit.full <- glm(formula0, family=binomial, data=dat)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fit.full)
```

```
##
## Call:
## glm(formula = formula0, family = binomial, data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1568  -1.0990   0.4168   0.9056   1.4206
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.8739921  0.6728320  -1.299  0.19395
## age                  0.0185989  0.0063716   2.919  0.00351 **
## factor(gender)Male   0.0316994  0.2296822   0.138  0.89023
## TB                   0.0085071  0.0828948   0.103  0.91826
## DB                   0.5074010  0.2377033   2.135  0.03279 *
## alkphos              0.0013148  0.0008117   1.620  0.10528
## sgpt                 0.0099959  0.0048651   2.055  0.03992 *
## sgot                 0.0032620  0.0031907   1.022  0.30661
## alb                  0.0006393  0.1851545   0.003  0.99725
## AGratio            -0.4370765  0.4953257  -0.882  0.37756
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 698.37  on 582  degrees of freedom
## Residual deviance: 579.65  on 573  degrees of freedom
## AIC: 599.65
##
## Number of Fisher Scoring iterations: 7
```

```
names(summary(fit.full))
```

```
##  [1] "call"           "terms"          "family"         "deviance"
##  [5] "aic"            "contrasts"      "df.residual"    "null.deviance"
##  [9] "df.null"        "iter"           "deviance.resid" "coefficients"
## [13] "aliased"        "dispersion"     "df"             "cov.unscaled"
## [17] "cov.scaled"
```

```
print("BIC")
```

```
## [1] "BIC"
```

```
BIC(fit.full)
```

```
## [1] 643.3317
```

Comment We see from the output that the variables age, DB and sgpt are statistically significant since their p-values are less than 0.05. Also, the BIC is 643.3317.

(b) Then select your ‘best’ model stepwise selection at the aid of BIC. This can be done by choosing direction=“both” and k=log(n) in the step() function. Call the resultant model as fit.step.

```
#Stepwise Variable Selection (SVS)
set.seed(123)
fit.step <- suppressWarnings(step(fit.full, direction = c("both"),
                                  k = log(NROW(dat)), trace = F))
summary(fit.step)
```

```
##
## Call:
## glm(formula = liver ~ age + DB + sgpt, family = binomial, data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0299  -1.1238   0.4760   0.9075   1.3869
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.122642   0.326753  -3.436 0.000591 ***
## age          0.019936   0.006123   3.256 0.001129 **
## DB           0.657824   0.175644   3.745 0.000180 ***
## sgpt         0.015096   0.003816   3.957  7.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 698.37  on 582  degrees of freedom
## Residual deviance: 586.96  on 579  degrees of freedom
## AIC: 594.96
##
## Number of Fisher Scoring iterations: 7
```

```
print("BIC")
```

```
## [1] "BIC"
```

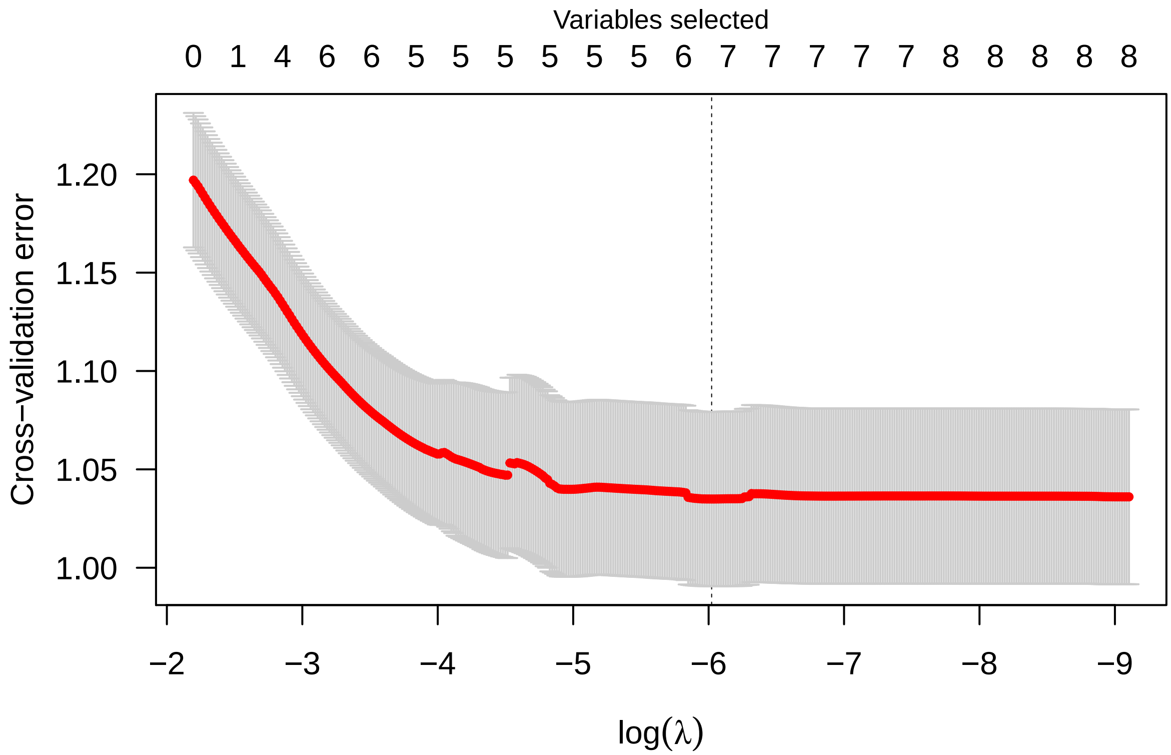
```
BIC(fit.step)
```

```
## [1] 612.4361
```

Comment From the output, its clearly that the variables age,DB and sgpt are statistically significant. Also, we observe that the BIC is 612.4361 is less than the previous BIC=643.3317.

(c) Next select your 'best' model with one of the regularization methods with different types of penalties, i.e., LASSO, SCAD, or MCP. Call the resultant model as fit.pen

```
#SCAD
set.seed(123)
library(ncvreg)
y <- dat$liver
X <- model.matrix(object=~ age + factor(gender) + TB + DB + alkphos + sgpt + sgot
                  + alb + AGratio, data=dat)
cvfit.SCAD <- cv.ncvreg(X=X,y=y, nfolds=5, family="binomial", penalty="SCAD",
                        lambda.min=.001, nlambda=400, eps=.01, max.iter=3000)
plot(cvfit.SCAD)
```



Comment We see from the plot that six variables are selected as important/relevant.

```
result.SCAD <- cvfit.SCAD$fit
beta.hat <- as.vector(result.SCAD$beta[-1, cvfit.SCAD$min])
cutoff <- 0
terms <- colnames(X)[abs(beta.hat) > cutoff]
print("Important Variables")
```

```
## [1] "Important Variables"
```

```
terms
```

```
## [1] "age"                 "factor(gender)Male" "DB"
## [4] "alkphos"             "sgpt"               "AGratio"
```

```
terms[2] <- c("factor(gender)")
formula.SCAD <- as.formula(paste(c("liver ~ 1", terms), collapse = " + "))
fit.pen <- glm(formula.SCAD, data = dat, family="binomial")
summary(fit.pen)
```

```
##
## Call:
## glm(formula = formula.SCAD, family = "binomial", data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2063  -1.1067   0.4251   0.9073   1.4116
##
```

```
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         -0.8641761  0.5817221  -1.486  0.13740    
## age                  0.0186006  0.0062472   2.977  0.00291 ** 
## factor(gender)Male   0.0424693  0.2283790   0.186  0.85248    
## DB                   0.5623188  0.1729392   3.252  0.00115 ** 
## alkphos              0.0013207  0.0008042   1.642  0.10054    
## sgpt                 0.0134185  0.0038492   3.486  0.00049 ***
## AGratio             -0.4431380  0.3627138  -1.222  0.22181    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 698.37  on 582  degrees of freedom
## Residual deviance: 580.84  on 576  degrees of freedom
## AIC: 594.84
## 
## Number of Fisher Scoring iterations: 7
```

```r
print("BIC")
```

```
## [1] "BIC"
```

```r
BIC(fit.pen)
```

```
## [1] 625.4182
```

Comment From the output, we see that only the variables age, DB and sgpt were found to be statistically significant. However, SCAD approach found the variables gender, alkphos, and AGratio to be important variables even though they are not statistically significant in the model. Also, the BIC is 625.4182.

Model Comparison

In order to make a resolution on the final model, let us compare the three models in terms of the area under the ROC curve (AUC) or the C-statistics. In order to have a more ‘honest’ comparison, we will compare them on the basis of their predicted probabilities after cross- validation. a) Compute the jackknife predicted probabilities from every model.

```r
set.seed(123)
n <- NROW(dat)
pop.jk <- matrix(rep(0, 3*n), ncol = 3)
model.names <- c("fit.full", "fit.step", "fit.pen")
for (i in 1:n){
  fit1.i <- suppressWarnings(glm(formula(fit.full), data=dat[-i,],
                                 family = "binomial"))
  fit2.i <- suppressWarnings(glm(formula(fit.step), data=dat[-i,],
                                 family = "binomial"))
  fit3.i <- suppressWarnings(glm(formula(fit.pen), data=dat[-i,],
                                 family = "binomial"))
```

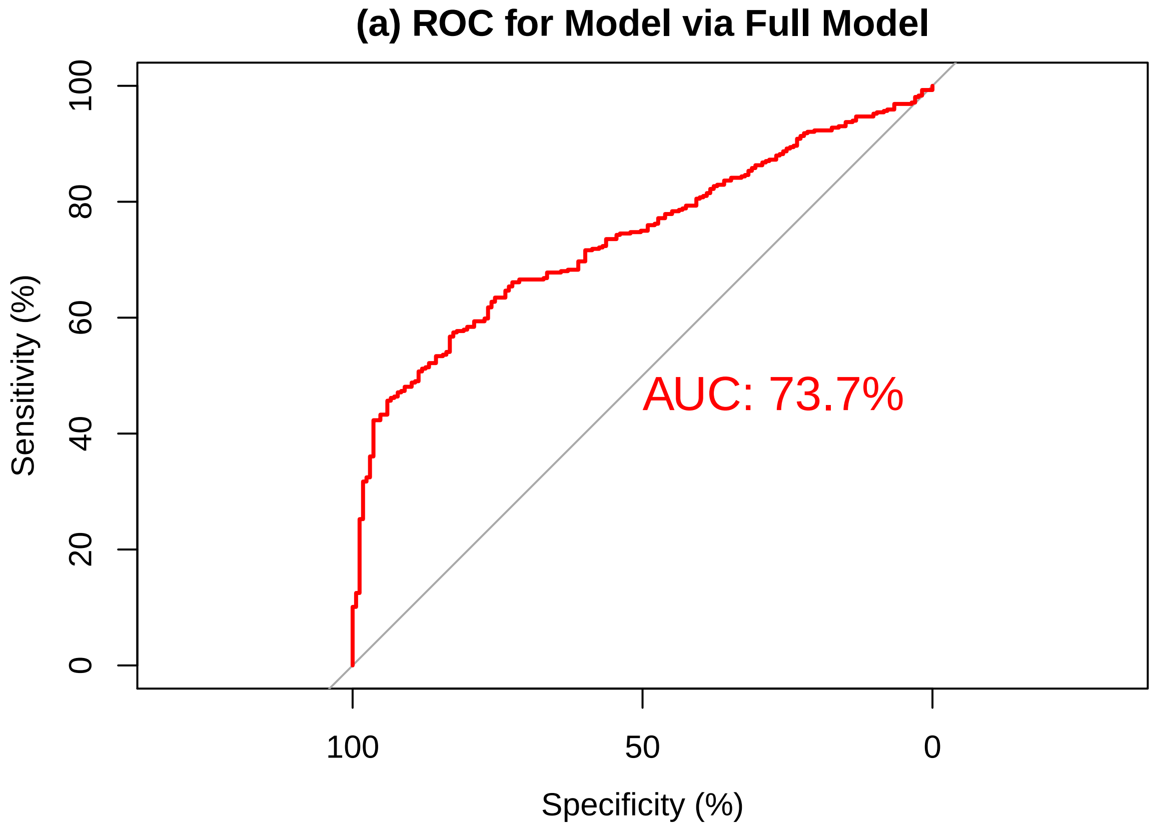
```
    pop.jk[i,1] <- predict(fit1.i, newdata=dat[i,], type="response")
    pop.jk[i,2] <- predict(fit2.i, newdata=dat[i,], type="response")
    pop.jk[i,3] <- predict(fit3.i, newdata=dat[i,], type="response")
  }

p.jk.fit.full <- as.vector(pop.jk[,1])
p.jk.fit.step <- as.vector(pop.jk[,2])
p.jk.fit.pen <- as.vector(pop.jk[,3])
```

(b) Plot their ROC curves and find their AUC values. Which model provides the largest AUC?

```
set.seed(123)
suppressPackageStartupMessages(library(pROC))
y <- dat$liver
roc.full <- plot.roc(y, p.jk.fit.full,  ylim=c(0, 100),
    main="(a) ROC for Model via Full Model", percent=TRUE,
    print.auc=TRUE, print.auc.cex=1.5, col="red")
```

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

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

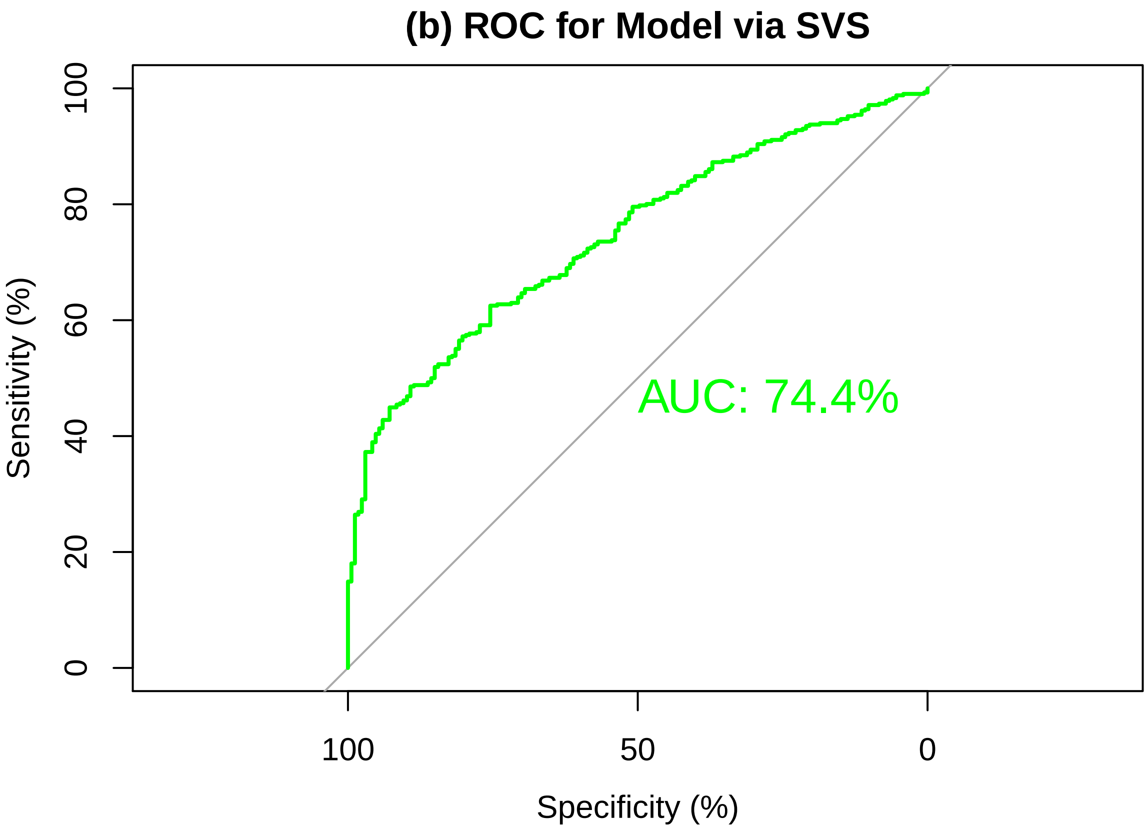


Comment The area under the ROC is AUC=73.7%

```
roc.step <- plot.roc(y, p.jk.fit.step, ylim=c(0, 100),
    main="(b) ROC for Model via SVS", percent=TRUE,
    print.auc=TRUE, print.auc.cex=1.5, col="green")
```

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

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

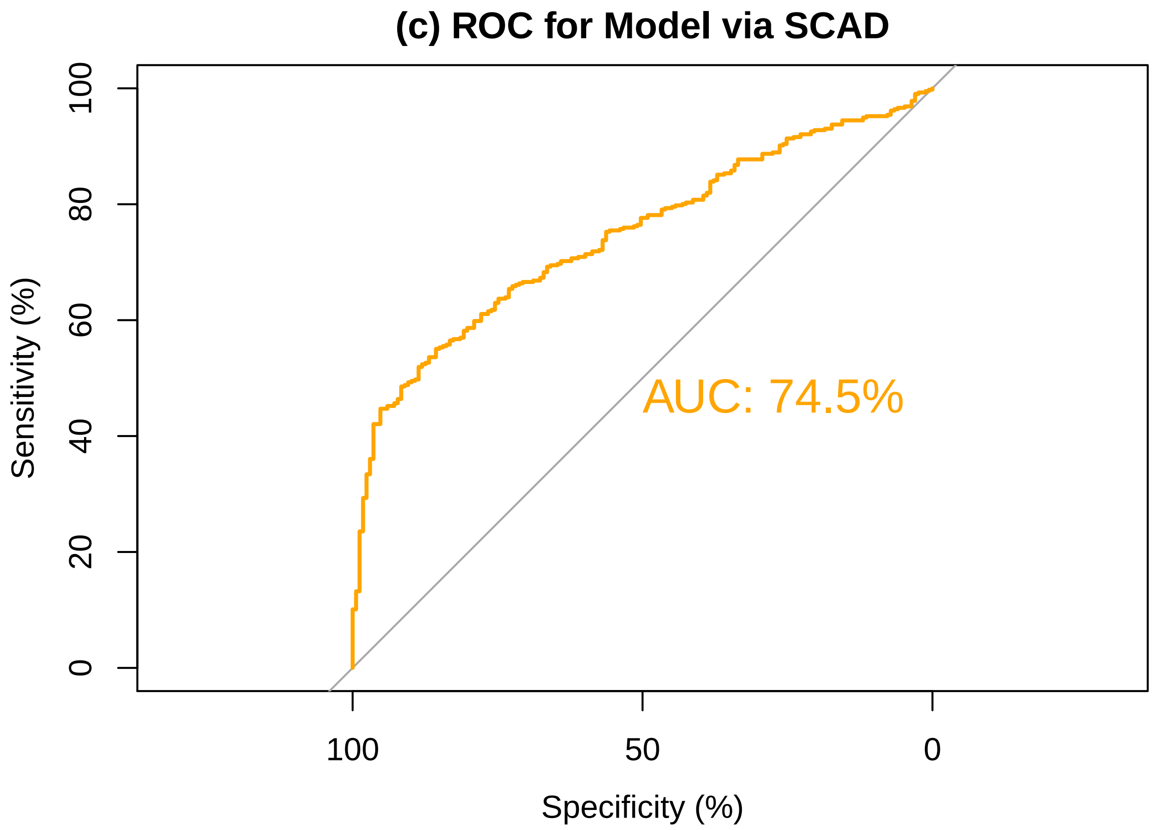


Comment The area under the ROC is AUC=74.4%

```
roc.SCAD <- plot.roc(y, p.jk.fit.pen, ylim=c(0, 100),
    main="(c) ROC for Model via SCAD", percent=TRUE,
    print.auc=TRUE, print.auc.cex=1.5, col="orange")
```

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

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



Comment The area under the ROC is AUC=74.5%

Conclusion We see from the above three curves that the model via SCAD has the largest AUC of 74.5%. Hence, we select this as our final model.

Final Best Logistic Model/Confidence intervals/Odd ratios

Finally, present your final best logistic model and output the 95% confidence intervals for coefficients Bj 's as well as their associated odds ratio (i.e., exp(Bj)). Interpret the results within the liver disease diagnostic context.

```
set.seed(123)
summary(fit.pen) # summary of the final best logistic model
```

```
##
## Call:
## glm(formula = formula.SCAD, family = "binomial", data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2063  -1.1067   0.4251   0.9073   1.4116
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.8641761  0.5817221  -1.486  0.13740
## age                   0.0186006  0.0062472   2.977  0.00291 **
## factor(gender)Male  0.0424693  0.2283790   0.186  0.85248    
## DB                  0.5623188  0.1729392   3.252  0.00115 ** 
## alkphos             0.0013207  0.0008042   1.642  0.10054    
## sgpt                0.0134185  0.0038492   3.486  0.00049 ***
## AGratio            -0.4431380  0.3627138  -1.222  0.22181    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 698.37  on 582  degrees of freedom
## Residual deviance: 580.84  on 576  degrees of freedom
## AIC: 594.84
##
## Number of Fisher Scoring iterations: 7
```

```
# CONFIDENCE INTERVAL FOR BETA’S
library(MASS)
ci <- suppressWarnings(confint(fit.pen, level = 0.95))
```

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

```
print("Confidence Interval for Beta’s")
```

```
## [1] "Confidence Interval for Beta’s"
```

```
ci
```

```
##                            2.5 %      97.5 %
## (Intercept)        -2.022747e+00 0.261941916
## age                 6.458542e-03 0.030990415
## factor(gender)Male -4.091796e-01 0.487365951
## DB                  2.755504e-01 0.946789837
## alkphos            -6.078321e-05 0.003084784
## sgpt                6.608318e-03 0.021670064
## AGratio            -1.150959e+00 0.272504234
```

```
print("Odds Ratio")
```

```
## [1] "Odds Ratio"
```

```
exp(fit.pen$coefficients)
```

```
##        (Intercept)                age factor(gender)Male                 DB
##          0.4213986          1.0187747          1.0433841          1.7547366
##            alkphos               sgpt            AGratio
##          1.0013216          1.0135089          0.6420186
```

```
print("95% CI of Odds Ratio")
```

```
## [1] "95% CI of Odds Ratio"
```

```
exp(ci)
```

```
##                         2.5 %   97.5 %
## (Intercept)        0.1322915 1.299451
## age                1.0064794 1.031476
## factor(gender)Male 0.6641949 1.628022
## DB                 1.3172554 2.577422
## alkphos            0.9999392 1.003090
## sgpt               1.0066302 1.021907
## AGratio            0.3163333 1.313249
```

Comment First, we see that all the values of odds Ratio for the variables lie within the 95% confidence interval of odds ratios which means that the variables age, gender(Male), DB, alkphos,sgpt and AGratio are factors for liver disease.So, the liver disease is likely to occur. Also, we see that the confidence intervals for Betas with respect to the variables age, DB and sgpt are positive which implies positive relationship.