Project VI

Isaiah Thompson Ocansey

4/29/2022

```
rm(list=ls(all=TRUE))

#loading the data
dat0<-read.csv("C:/Users/thomo/OneDrive/Desktop/Data Mining/hcvdat0.csv")
dim(dat0); head(dat0)

## [1] 615  14

##   X      Category Age Sex  ALB  ALP  ALT  AST  BIL   CHE  CHOL CREA  GGT
PROT
## 1 1 0=Blood Donor  32   m 38.5 52.5  7.7 22.1  7.5  6.93 3.23  106 12.1
69.0
## 2 2 0=Blood Donor  32   m 38.5 70.3 18.0 24.7  3.9 11.17 4.80   74 15.6
76.5
## 3 3 0=Blood Donor  32   m 46.9 74.7 36.2 52.6  6.1  8.84 5.20   86 33.2
79.3
## 4 4 0=Blood Donor  32   m 43.2 52.0 30.6 22.6 18.9  7.33 4.74   80 33.8
75.7
## 5 5 0=Blood Donor  32   m 39.2 74.1 32.6 24.8  9.6  9.15 4.32   76 29.9
68.7
## 6 6 0=Blood Donor  32   m 41.6 43.3 18.5 19.7 12.3  9.92 6.05  111 91.0
74.0
```

The data has 14 variables and 615 observations

1) Data Cleaning: Prepare the data.
(a) Remove the first ID column

```
dat0=dat0[,-1]
head(dat0);dim(dat0)

##        Category Age Sex  ALB  ALP  ALT  AST  BIL   CHE  CHOL CREA  GGT PROT
## 1 0=Blood Donor  32   m 38.5 52.5  7.7 22.1  7.5  6.93 3.23  106 12.1 69.0
## 2 0=Blood Donor  32   m 38.5 70.3 18.0 24.7  3.9 11.17 4.80   74 15.6 76.5
## 3 0=Blood Donor  32   m 46.9 74.7 36.2 52.6  6.1  8.84 5.20   86 33.2 79.3
## 4 0=Blood Donor  32   m 43.2 52.0 30.6 22.6 18.9  7.33 4.74   80 33.8 75.7
## 5 0=Blood Donor  32   m 39.2 74.1 32.6 24.8  9.6  9.15 4.32   76 29.9 68.7
## 6 0=Blood Donor  32   m 41.6 43.3 18.5 19.7 12.3  9.92 6.05  111 91.0 74.0

## [1] 615  13
```

After removing the first ID variable, we have 615 observations and 13 variables

(c) What is the proportion of subjects who were diagnosed as Hepatitis C? Does it present an unbalanced classification problem?

```
prop <- data.frame(table(dat0$Category=="1=Hepatitis"))
prop$Proportion <- (prop$Freq/sum(prop$Freq))*100
names(prop) <- c("Hepatitis","Frequency","Proportion")
prop

##   Hepatitis Frequency Proportion
## 1     FALSE       591  96.097561
## 2      TRUE        24   3.902439
```

The proportion of subjects diagnosed as Hepatitis C is 3.9:96.097 and it represent an unbalanced classification problem.

(b) The target variable Category have several diagnosis values: '0=Blood Donor', '0s=suspect Blood Donor', '1=Hepatitis', '2=Fibrosis', '3=Cirrhosis'. Change it to binary by setting the first two categories as 0 and others as 1.

```
dat0$Category <- ifelse(dat0$Category=="0=Blood
Donor"|dat0$Category=="0s=suspect Blood Donor", 0, 1);

table(dat0$Category)

##
##   0   1
## 540  75
```

After converting the response variable to a binary variable, we have 540 0s and 75 1s.

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

```
# obtaining the missing percentage of each variable
missing_rate <- data.frame(stringsAsFactors = F)
nr <- NROW(dat0)
nc <- NCOL(dat0)
Var_name <- variable.names(dat0)
for (i in 1:nc) {
  na <- sum(is.na(dat0[,i]))
  na_rate <- (na/nr)*100
  result <- list(Variable = Var_name[i], Number_Missing = na,
                 Missing_Rate = na_rate)
  missing_rate <- rbind(missing_rate, result, stringsAsFactors = F)
}
missing_rate

##    Variable Number_Missing Missing_Rate
## 1  Category              0    0.0000000
## 2       Age              0    0.0000000
## 3       Sex              0    0.0000000
## 4       ALB              1    0.1626016
## 5       ALP             18    2.9268293
## 6       ALT              1    0.1626016
## 7       AST              0    0.0000000
## 8       BIL              0    0.0000000
## 9       CHE              0    0.0000000
## 10     CHOL             10    1.6260163
## 11     CREA              0    0.0000000
## 12      GGT              0    0.0000000
## 13     PROT              1    0.1626016
```

Yes, There are missing values and they are shown above; the variable ALP has 18 missing values, ALB has 1,ALT has 1,CHOL has 10 etc.

```
set.seed(125)
suppressPackageStartupMessages(library(mice))
data_imputed <- mice(dat0, printFlag = F)

## Warning: Number of logged events: 1

dat <- complete(data_imputed, 1)
dat <- as.data.frame(dat)
rm(data_imputed)
dim(dat);head(dat)

## [1] 615  13

##   Category Age Sex  ALB  ALP  ALT  AST  BIL   CHE CHOL CREA  GGT PROT
## 1        0  32   m 38.5 52.5  7.7 22.1  7.5  6.93 3.23  106 12.1 69.0
## 2        0  32   m 38.5 70.3 18.0 24.7  3.9 11.17 4.80   74 15.6 76.5
## 3        0  32   m 46.9 74.7 36.2 52.6  6.1  8.84 5.20   86 33.2 79.3
## 4        0  32   m 43.2 52.0 30.6 22.6 18.9  7.33 4.74   80 33.8 75.7
## 5        0  32   m 39.2 74.1 32.6 24.8  9.6  9.15 4.32   76 29.9 68.7
## 6        0  32   m 41.6 43.3 18.5 19.7 12.3  9.92 6.05  111 91.0 74.0
```

The missing values have been computed using the package mice as shown above.

2) EDA and Variable Screening: We first explore the data with some simple statistical summary.

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

```
str(dat)

## 'data.frame':    615 obs. of  13 variables:
##  $ Category: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Age     : int  32 32 32 32 32 32 32 32 32 32 ...
##  $ Sex     : chr  "m" "m" "m" "m" ...
##  $ ALB     : num  38.5 38.5 46.9 43.2 39.2 41.6 46.3 42.2 50.9 42.4 ...
##  $ ALP     : num  52.5 70.3 74.7 52 74.1 43.3 41.3 41.9 65.5 86.3 ...
##  $ ALT     : num  7.7 18 36.2 30.6 32.6 18.5 17.5 35.8 23.2 20.3 ...
##  $ AST     : num  22.1 24.7 52.6 22.6 24.8 19.7 17.8 31.1 21.2 20 ...
##  $ BIL     : num  7.5 3.9 6.1 18.9 9.6 12.3 8.5 16.1 6.9 35.2 ...
##  $ CHE     : num  6.93 11.17 8.84 7.33 9.15 ...
##  $ CHOL    : num  3.23 4.8 5.2 4.74 4.32 6.05 4.79 4.6 4.1 4.45 ...
##  $ CREA    : num  106 74 86 80 76 111 70 109 83 81 ...
##  $ GGT     : num  12.1 15.6 33.2 33.8 29.9 91 16.9 21.5 13.7 15.9 ...
##  $ PROT    : num  69 76.5 79.3 75.7 68.7 74 74.5 67.1 71.3 69.9 ...
```

From the above, we have one integer variable(age),one categorical variable(Sex) and the rest of the variables are continuous

(b) For each categorical predictor, use χ2 test of independent to assess its association with the binary response; For other types of predictors, use either two-sample t test or the nonparametric Wilcoxon rank-sum test. Output the p-value for each variable. Alternatively, you may use simple logistic regression for this purpose.

Test of Independent and Two Sample T-Test

```
cond <- dat$Category == 1|dat$Category==0
cond1 <- as.vector(which(sapply(dat, is.numeric), arr.ind = T))
print("Test of Normality of the response variable Category")

## [1] "Test of Normality of the response variable Category"

sapply(dat[cond, cond1], shapiro.test)

##           Category                      Age
## statistic 0.3821251                     0.9841004
## p.value   2.990891e-41                  3.069652e-06
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                      "X[[i]]"
##           ALB                           ALP
## statistic 0.9280893                     0.7692807
## p.value   1.387319e-16                  1.69207e-28
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                      "X[[i]]"
##           ALT                           AST
## statistic 0.5815027                     0.4705794
## p.value   6.520306e-36                  4.354491e-39
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                      "X[[i]]"
##           BIL                           CHE
## statistic 0.3122516                     0.9809562
## p.value   8.713313e-43                  3.53716e-07
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                      "X[[i]]"
##           CHOL                          CREA
## statistic 0.9882472                     0.2465916
## p.value   7.581731e-05                  4.075916e-44
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                      "X[[i]]"
##           GGT                           PROT
## statistic 0.4804216                     0.9461568
## p.value   7.898311e-39                  3.69445e-14
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                      "X[[i]]"
```

From the Shapiro-Wilk Normality test, the normality assumption is violated so I use the Wilcoxon rank-sum test

```
set.seed(125)
suppressPackageStartupMessages(library(car))
vars.nominal <- c("Sex")
cols.x <- 1:(NCOL(dat)-1)
xnames <- names(dat)[cols.x]
y <- dat$Category
OUT <- NULL
for (j in 1:length(cols.x)){
  x <- dat[, cols.x[j]]
  xname <- xnames[j]
  if (is.element(xname, vars.nominal)){
    tbl <- table(x, y)
    pvalue <- chisq.test(tbl)$p.value
  } else {
    # WILCOXON TEST
    pvalue <- wilcox.test(x~y, alternative="two.sided")$p.value
  }
  OUT <- rbind(OUT, cbind(xname=xname, pvalue=pvalue))
}
OUT <- as.data.frame(OUT, stringsAsFactors =F)
colnames(OUT) <- c("name", "pvalue")
OUT$pvalue <- round(as.numeric(OUT$pvalue), digits = 4)
OUT

##         name pvalue
## 1   Category 0.0000
## 2        Age 0.2368
## 3        Sex 0.0988
## 4        ALB 0.0006
## 5        ALP 0.0000
## 6        ALT 0.0028
## 7        AST 0.0000
## 8        BIL 0.0000
## 9        CHE 0.0001
## 10      CHOL 0.0000
## 11      CREA 0.0026
## 12       GGT 0.0000
```

It can be observed from the table above that there was no association between Sex,Age and being diagnosed of Hepatitis since their p-values are 0.09 and 0.236 respectively. However, it can be observed that there is an association between hepatitis and the rest of the predictor variables

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

```
#Variable screening
cond2 <- OUT$pvalue > 0.20
OUT[cond2,]

##   name pvalue
## 2  Age 0.2368
```

3) Variable Selection:

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

```
#Excluding age
#Full model.
set.seed(125)
formula0 <- Category~  factor(Sex) + ALB + ALP + ALT + AST +BIL+ CHE +CHOL+
CREA + GGT+ PROT

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
## -4.8285  -0.1842  -0.0908  -0.0417   3.7328
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -14.316537   3.610514  -3.965 7.33e-05 ***
## factor(Sex)m  -1.021514   0.593162  -1.722  0.08504 .
## ALB           -0.094829   0.061854  -1.533  0.12525
## ALP           -0.064959   0.012065  -5.384 7.28e-08 ***
## ALT           -0.023600   0.009052  -2.607  0.00913 **
## AST            0.092392   0.019445   4.752 2.02e-06 ***
## BIL            0.073485   0.027682   2.655  0.00794 **
## CHE            0.139371   0.126348   1.103  0.27000
## CHOL          -0.842369   0.267093  -3.154  0.00161 **
## CREA           0.023500   0.005223   4.500 6.81e-06 ***
## GGT            0.031268   0.006293   4.969 6.73e-07 ***
## PROT           0.229006   0.058829   3.893 9.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 456.08  on 614  degrees of freedom
## Residual deviance: 127.47  on 603  degrees of freedom
## AIC: 151.47
##
## Number of Fisher Scoring iterations: 8

print("BIC")

## [1] "BIC"

BIC(fit.full)

## [1] 204.5297
```

In the full model the variables;ALP,ALT,AST,BIL,CHOL,CREA,GGT and PROT are statistically significant.

(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

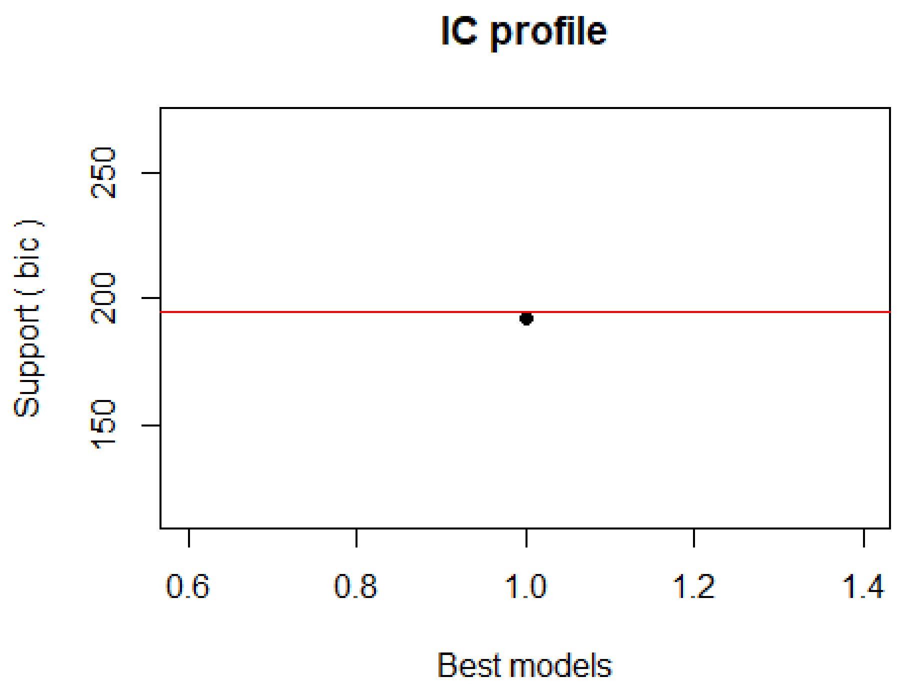
```
#Stepwise Variable Selection (SVS)
set.seed(125)
fit.step <- suppressWarnings(step(fit.full, direction = c("both"),
                                  k = log(NROW(dat)), trace = F))
summary(fit.step)

##
## Call:
## glm(formula = Category ~ ALP + ALT + AST + BIL + CHOL + CREA +
##     GGT + PROT, family = binomial, data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.2133  -0.2031  -0.1036  -0.0423   3.4275
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.661493   3.597607  -4.075 4.59e-05 ***
## ALP          -0.058786   0.011252  -5.225 1.75e-07 ***
## ALT          -0.026745   0.008892  -3.008  0.00263 **
## AST           0.101574   0.019055   5.331 9.79e-08 ***
## BIL           0.054455   0.025889   2.103  0.03544 *
## CHOL         -0.851239   0.241334  -3.527  0.00042 ***
## CREA          0.021526   0.005439   3.957 7.58e-05 ***
## GGT           0.029199   0.005803   5.032 4.86e-07 ***
## PROT          0.185642   0.046980   3.952 7.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 456.08  on 614  degrees of freedom
## Residual deviance: 133.77  on 606  degrees of freedom
## AIC: 151.77
##
## Number of Fisher Scoring iterations: 8

print("BIC")

## [1] "BIC"

BIC(fit.step)

## [1] 191.5631
```

the variables;ALP,ALT,AST,BIL,CHOL,CREA,GGT and PROT are statistically significant.

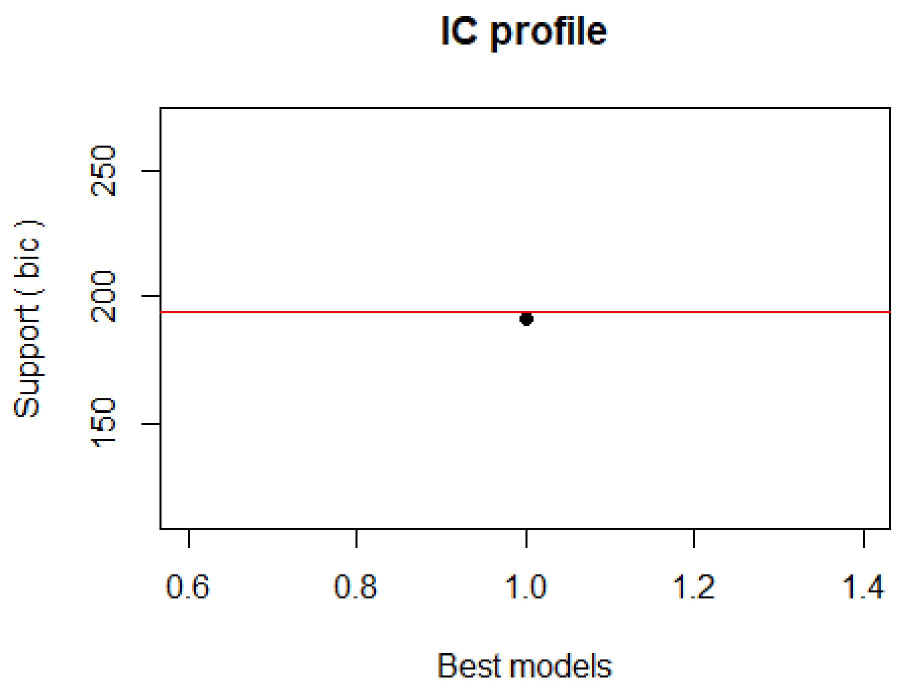
(c) Next, select your ‘best’ model with the best subset selection (BSS) based on minimum BIC. Name the resultant model as fit.bss.

```
library(rJava)
library(glmulti)

## Loading required package: leaps

##
## Attaching package: 'glmulti'

## The following object is masked from 'package:mice':
##
##     getfit

fitting <- suppressWarnings(glmulti(Category~  factor(Sex) + ALB + ALP + ALT
+ AST +BIL+ CHE +CHOL+ CREA + GGT+ PROT,  # CAN HANDLE CATEGORICAL PREDICTORS
cat
    data = dat, fitfunc = glm, family=binomial, intercept = TRUE,
    crit = bic, level = 1, method="g",
    confsetsize=1))  # SELECT ONLY ONE BEST MODEL

## Initialization...
## TASK: Genetic algorithm in the candidate set.
## Initialization...
## Algorithm started...
##
## After 10 generations:
## Best model: Category~1+ALP+ALT+AST+CHOL+CREA+GGT+PROT
## Crit= 192.16942367421
## Mean crit= 192.16942367421
```



```
## Change in best IC: -9807.83057632579 / Change in mean IC: -
9807.83057632579
##
## After 20 generations:
## Best model: Category~1+ALP+ALT+AST+BIL+CHOL+CREA+GGT+PROT
## Crit= 191.563103003137
## Mean crit= 191.563103003137
```

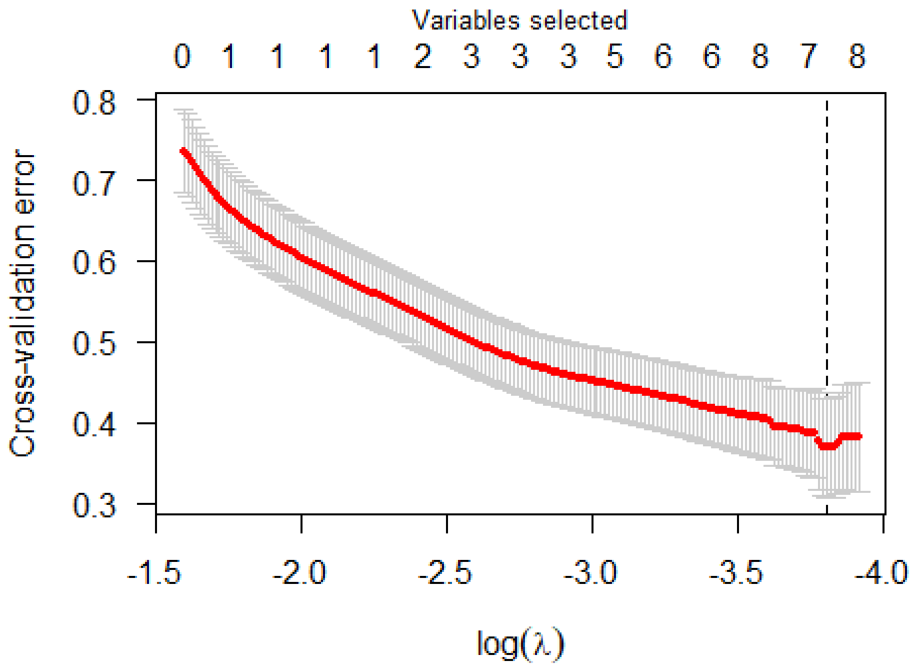
```
## Change in best IC: -0.606320671072524 / Change in mean IC: -
0.606320671072524
##
## After 30 generations:
## Best model: Category~1+ALP+ALT+AST+BIL+CHOL+CREA+GGT+PROT
## Crit= 191.563103003137
## Mean crit= 191.563103003137
```



```
## Change in best IC: 0 / Change in mean IC: 0
##
## After 40 generations:
## Best model: Category~1+ALP+ALT+AST+BIL+CHOL+CREA+GGT+PROT
## Crit= 191.563103003137
## Mean crit= 191.563103003137

## Change in best IC: 0 / Change in mean IC: 0
##
## After 50 generations:
## Best model: Category~1+ALP+ALT+AST+BIL+CHOL+CREA+GGT+PROT
## Crit= 191.563103003137
## Mean crit= 191.563103003137

## Change in best IC: 0 / Change in mean IC: 0
##
## After 60 generations:
## Best model: Category~1+ALP+ALT+AST+BIL+CHOL+CREA+GGT+PROT
## Crit= 191.563103003137
## Mean crit= 191.563103003137

## Change in best IC: 0 / Change in mean IC: 0
##
## After 70 generations:
## Best model: Category~1+ALP+ALT+AST+BIL+CHOL+CREA+GGT+PROT
## Crit= 191.563103003137
## Mean crit= 191.563103003137
## Improvements in best and average IC have bebingo en below the specified
goals.
## Algorithm is declared to have converged.
## Completed.

fit.bss <- attributes(fitting)$objects[[1]]
fit.bss$coef

##  (Intercept)          ALP          ALT          AST          BIL
CHOL
## -14.66149290  -0.05878610  -0.02674454   0.10157446   0.05445454  -
0.85123892
##         CREA          GGT         PROT
##   0.02152564   0.02919859   0.18564170

summary(fit.bss)

##
## Call:
## fitfunc(formula = as.formula(x), family = ..1, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.2133  -0.2031  -0.1036  -0.0423   3.4275
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.661493   3.597607  -4.075 4.59e-05 ***
## ALP          -0.058786   0.011252  -5.225 1.75e-07 ***
## ALT          -0.026745   0.008892  -3.008  0.00263 **
## AST           0.101574   0.019055   5.331 9.79e-08 ***
## BIL           0.054455   0.025889   2.103  0.03544 *
## CHOL         -0.851239   0.241334  -3.527  0.00042 ***
## CREA          0.021526   0.005439   3.957 7.58e-05 ***
## GGT           0.029199   0.005803   5.032 4.86e-07 ***
## PROT          0.185642   0.046980   3.952 7.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 456.08  on 614  degrees of freedom
## Residual deviance: 133.77  on 606  degrees of freedom
## AIC: 151.77
##
## Number of Fisher Scoring iterations: 8
```

(d) Finally select your 'best' model with ONE of the regularization methods with different types of penalties, e.g., LASSO, Adaptive LASSO, SCAD, MCP, or MIC. Call the resultant model as fit.pen

```
#SCAD
set.seed(125)
library(ncvreg)
X <- model.matrix(object=~  factor(Sex) + ALB + ALP + ALT + AST +BIL+ CHE
+CHOL+ CREA + GGT+ PROT, data=dat)
y <- dat$Category
cvfit.SCAD <- cv.ncvreg(X=X,y=y, nfolds=5, family="binomial", penalty="SCAD",
                        lambda.min=.001, nlambda=500, eps=.01, max.iter=5000)

## Warning in ncvreg(X = X, y = y, ...): Maximum number of iterations reached

plot(cvfit.SCAD)
```



The plot suggests that there are 8 important variables

```
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] "ALP"   "ALT"   "AST"   "BIL"   "CHOL" "CREA" "GGT"   "PROT"
```

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

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

summary(fit.pen)

##
## Call:
## glm(formula = formula.SCAD, family = "binomial", data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9294  -0.2005  -0.1045  -0.0447   3.6101
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -12.416028   3.431183  -3.619 0.000296 ***
## ALP           -0.063816   0.010787  -5.916 3.29e-09 ***
## factor(Sex)m  -1.098052   0.515561  -2.130 0.033187 *
## AST            0.065898   0.011963   5.508 3.62e-08 ***
## BIL            0.077528   0.024439   3.172 0.001512 **
## CHOL          -0.796793   0.233938  -3.406 0.000659 ***
## CREA           0.022573   0.005042   4.477 7.58e-06 ***
## GGT            0.032306   0.006001   5.384 7.30e-08 ***
## PROT           0.163170   0.044017   3.707 0.000210 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 456.08  on 614  degrees of freedom
## Residual deviance: 139.64  on 606  degrees of freedom
## AIC: 157.64
##
## Number of Fisher Scoring iterations: 8

BIC(fit.pen)

## [1] 197.438
```

4) In order to make a resolution on the final model. Let's compare the four models in terms of the area under the ROC curve (AUC) or the C-statistics. In order to have a more 'honest' comparison, we shall compare them on the basis of their predicted probabilities after crossvalidation.

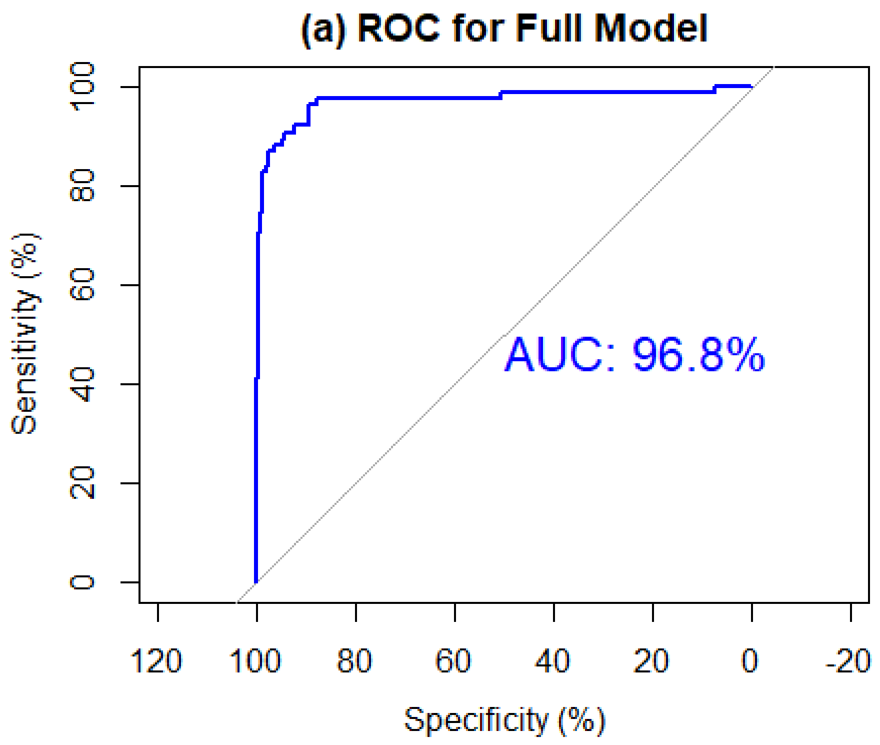
(a) Compute the jackknife predicted probabilities from every model.

Model Comparison

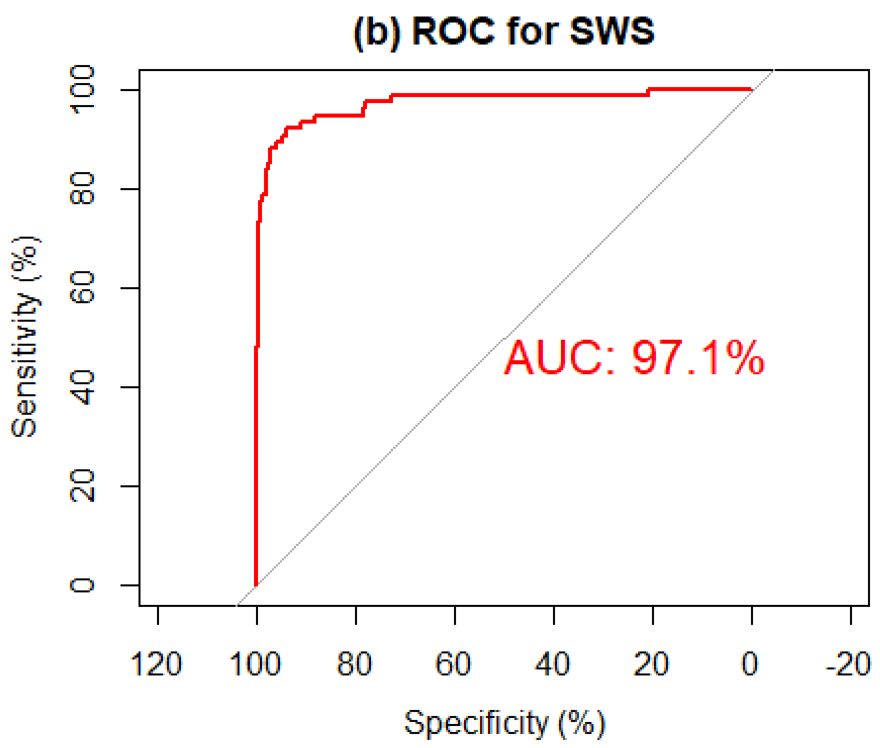
```
set.seed(125)
n <- NROW(dat)
pop.jk <- matrix(rep(0, 4*n), ncol = 4)
model.names <- c("fit.full", "fit.step", "fit.pen")
  for (i in 1:n){
    fit1.i <- suppressWarnings(glm(formula(fit.full), data=dat[-i,],
                                      family = "binomial"))
        fit2.i <- suppressWarnings(glm(formula(fit.step), data=dat[-i,],
                                      family = "binomial"))
            fit3.i <- suppressWarnings(glm(formula(fit.pen), data=dat[-i,],
                                      family = "binomial"))
            fit4.i <- suppressWarnings(glm(formula(fit.bss), 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")
    pop.jk[i,4] <- predict(fit4.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])
p.jk.fit.best <- as.vector(pop.jk[,4])
```

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

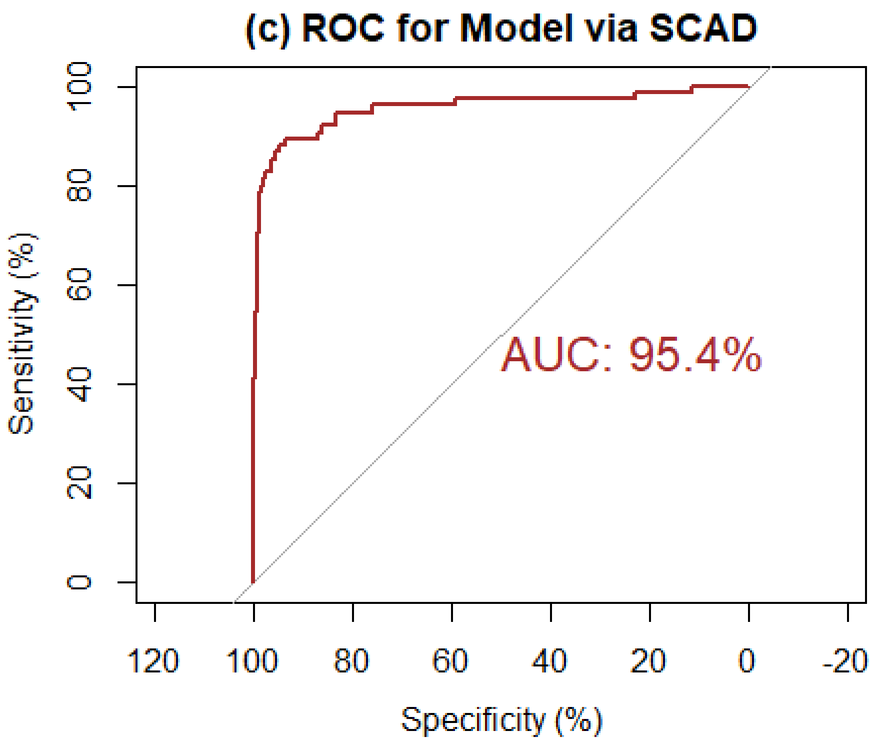
```
set.seed(123)
suppressPackageStartupMessages(library(pROC))
y <- dat$Category
roc.full <- plot.roc(y, p.jk.fit.full,  ylim=c(0, 100),
    main="(a) ROC for Full Model", percent=TRUE,
    print.auc=TRUE, print.auc.cex=1.5, col="blue")

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

## Setting direction: controls < cases
```



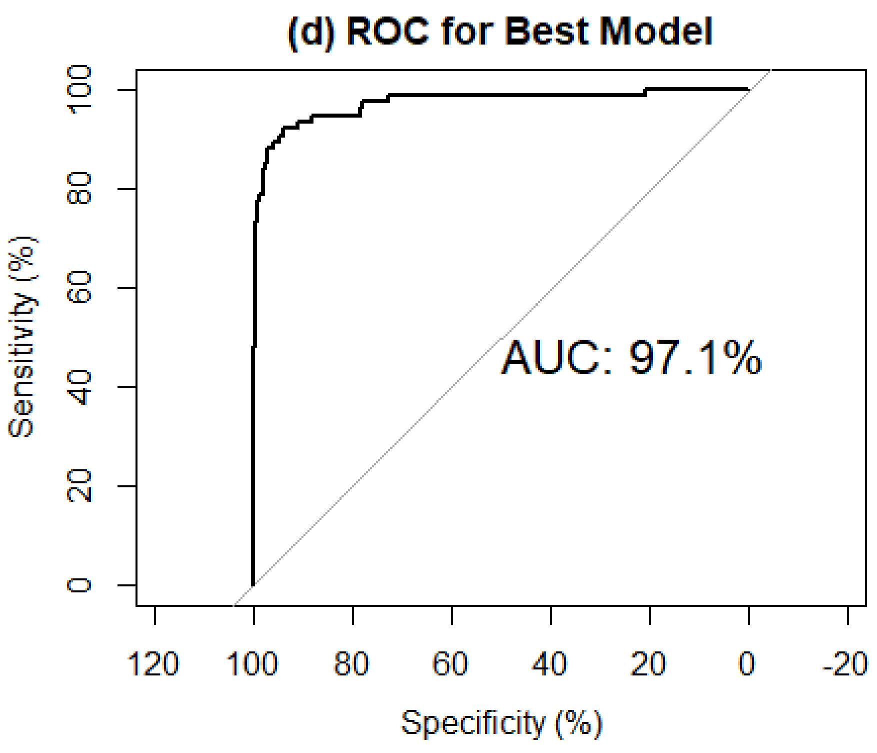
```
roc.step <- plot.roc(y, p.jk.fit.step, ylim=c(0, 100),
    main="(b) ROC for SWS", percent=TRUE,
    print.auc=TRUE, print.auc.cex=1.5, col="red")

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

## Setting direction: controls < cases
```



```
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="brown")

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

## Setting direction: controls < cases
```



```
set.seed(123)
suppressPackageStartupMessages(library(pROC))
y <- dat$Category
roc.full <- plot.roc(y, p.jk.fit.best,  ylim=c(0, 100),
    main="(d) ROC for Best Model", percent=TRUE,
    print.auc=TRUE, print.auc.cex=1.5, col="black")

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

## Setting direction: controls < cases
```



From the above AUC, The stepwise selection and the best subset selection models have the higher area under curve of 97.1%.

5) Finally, present your final best logistic model and output the 95% confidence intervals for coefficients βj ’s, as well as their associated odds ratio (i.e., exp(βj)). Interpret the results within the problem context.

```
set.seed(125)
summary(fit.step)

##
## Call:
## glm(formula = Category ~ ALP + ALT + AST + BIL + CHOL + CREA +
##     GGT + PROT, family = binomial, data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.2133  -0.2031  -0.1036  -0.0423   3.4275
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.661493   3.597607  -4.075 4.59e-05 ***
## ALP          -0.058786   0.011252  -5.225 1.75e-07 ***
## ALT          -0.026745   0.008892  -3.008  0.00263 **
## AST           0.101574   0.019055   5.331 9.79e-08 ***
## BIL           0.054455   0.025889   2.103  0.03544 *
## CHOL         -0.851239   0.241334  -3.527  0.00042 ***
## CREA          0.021526   0.005439   3.957 7.58e-05 ***
## GGT           0.029199   0.005803   5.032 4.86e-07 ***
## PROT          0.185642   0.046980   3.952 7.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 456.08  on 614  degrees of freedom
## Residual deviance: 133.77  on 606  degrees of freedom
## AIC: 151.77
##
## Number of Fisher Scoring iterations: 8
```

```
# CONFIDENCE INTERVAL FOR BETA'S
library(MASS)
confinterval <- suppressWarnings(confint(fit.step, level = 0.95))
```

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

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

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

```
confinterval
```

```
##                     2.5 %       97.5 %
## (Intercept) -22.564593645 -8.259516891
## ALP          -0.082064752 -0.037009577
## ALT          -0.044969106 -0.009962774
## AST           0.066186181  0.140985815
## BIL           0.008965227  0.104638033
## CHOL         -1.351181964 -0.401130568
## CREA          0.011555561  0.033208629
## GGT           0.018693130  0.041780081
## PROT          0.101322219  0.288284410
```

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

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

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

```
##  (Intercept)          ALP          ALT          AST          BIL 
CHOL 
## 4.291356e-07 9.429084e-01 9.736099e-01 1.106912e+00 1.055964e+00 
4.268857e-01 
##         CREA          GGT         PROT 
## 1.021759e+00 1.029629e+00 1.203991e+00
```

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

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

```
exp(confinterval)
```

```
##                    2.5 %      97.5 %
## (Intercept) 1.586067e-10 0.000258784
## ALP         9.212123e-01 0.963666907
## ALT         9.560270e-01 0.990086690
## AST         1.068426e+00 1.151408315
## BIL         1.009006e+00 1.110308643
## CHOL        2.589340e-01 0.669562632
## CREA        1.011623e+00 1.033766190
## GGT         1.018869e+00 1.042665151
## PROT        1.106633e+00 1.334136692
```

It is observed that a unit change in ALP, ALT,AST,BIL,CHOL,CREA,GGT and PROT will bring about 4.291356e-07, 9.429084e-01, 9.736099e-01 ,1.106912e+00, 1.055964e+00, 4.268857e-01, 1.021759e+00, 1.029629e+00, 1.203991e+00 respective increase in the odds of being diagnosed as Hepatitis subject.