Project 6

2022-11-27

```r
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)
```

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

```r
head(ILPD)
```

```
##   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
```

```r
nrows = dim(ILPD)[1]
ncols = dim(ILPD)[2]
set.seed(1)
```

1. Data Cleaning

(a)

```r
100*table(ILPD$liver)/nrows
```

```
##
##        1        2
## 71.35506 28.64494
```

About 71% of individuals in the data set are diagnosed with liver disease while 29% are not. The data set is therefore very unbalanced—especially considering the prevalence of liver disease in India is much lower. According to a large-scale study conducted by Mukherjee and colleagues from 2010 to 2013, out of 20,701,383 patients, 266,621 were diagnosed with liver disease (Mukherjee, 2017). That is a pre)valence of about 1.28%.

(b)

```r
for (col in colnames(ILPD)) {
  print(col)
  print(sum(is.na(ILPD[,col])))
}
```

```
## [1] "age"
## [1] 0
## [1] "gender"
## [1] 0
## [1] "TB"
## [1] 0
## [1] "DB"
## [1] 0
## [1] "alkphos"
## [1] 0
## [1] "sgpt"
## [1] 0
## [1] "sgot"
## [1] 0
## [1] "TP"
## [1] 0
## [1] "alb"
## [1] 0
## [1] "AGratio"
## [1] 4
## [1] "liver"
## [1] 0
```

```r
mean_AGratio = mean(ILPD$AGratio, na.rm = TRUE)
for (i in c(1:nrows)) {
  if (is.na(ILPD[i, "AGratio"])) {
    ILPD[i, "AGratio"] = mean_AGratio
  }
}
```

After counting all missing values per variable, there were only four missing in the AGratio column. Because AGratio is a float variable between 0 and 1, I averaged all non-missing values in the column to impute the missing values with the average.

2. EDA and Variable Selection

(a)

Age is a continuous variable though we truncate the fractional part. It is therefore a count variable (counting the number of years a patient has been alive).

Gender is a categorical variable.

TB stands for total bilirubin and is a continuous variable.

DB stands for direct bilirubin and is a continuous variable.

Alkphos stands for Alkaline Phosphotase and is a measure enzymatic activity. Though enzymatic activity is technically continuous, in this study, it is presented as a count variable.

Sgpt stands for Alamine Aminotransferase and is a measure enzymatic activity. Though enzymatic activity is technically continuous, in this study, it is presented as a count variable.

Sgot stands for Aspartate Aminotransferase and is a measure enzymatic activity. Though enzymatic activity is technically continuous, in this study, it is presented as a count variable.

TP stands for total proteins and is a continuous variable.

ALB stands for albumin and is a continuous variable.

A/G ratio measures the albumin and globulin ratio and is a continuous variable.

Of course, liver is the response variable and is categorical.

Thus we have two categorical variables, four count variables, and five continuous variables.

```
ILPD$gender = as.factor(ILPD$gender)
```

(b)

```
library(car)
```

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

```
vars.nominal = c("gender")
cols.x = 1:(NCOL(ILPD)-1)
xnames = names(ILPD)[cols.x]
y = ILPD$liver
OUT = NULL
for (j in 1:length(cols.x)){
  x = ILPD[, cols.x[j]]
  xname = xnames[j]

  if (is.element(xname, vars.nominal)){
    tbl = table(x, y)
    pvalue = chisq.test(tbl)$p.value
  }

  else {
    # TWO-SAMPLE t TEST
    pvalue.equal.var <- (leveneTest(x~factor(y))$"Pr(>F)")[1]
    equal.var <- ifelse(pvalue.equal.var <= 0.05, FALSE, TRUE)
    pvalue <- t.test(x~y, alternative="two.sided",
    var.equal=equal.var)$p.value
  }

  OUT = rbind(OUT, cbind(xname=xname, pvalue=pvalue))
}# colnames(OUT) = c("name", "pvalue")
OUT
```

```
##       xname     pvalue
##  [1,] "age"     "0.000884063155626139"
##  [2,] "gender"  "0.0596658468577747"
##  [3,] "TB"      "4.91200919556184e-16"
##  [4,] "DB"      "2.26995324945029e-19"
##  [5,] "alkphos" "1.08124966726932e-08"
##  [6,] "sgpt"    "1.18047797924202e-09"
##  [7,] "sgot"    "1.40944977692876e-08"
##  [8,] "TP"      "0.398819127523851"
##  [9,] "alb"     "9.07436084295548e-05"
## [10,] "AGratio" "8.25114350471533e-05"
```

```
OUT = as.data.frame(OUT)
colnames(OUT) = c("name", "pvalue")
OUT
```

```
##       name                  pvalue
## 1       age 0.000884063155626139
## 2    gender   0.0596658468577747
## 3        TB 4.91200919556184e-16
## 4        DB 2.26995324945029e-19
## 5   alkphos 1.08124966726932e-08
## 6      sgpt 1.18047797924202e-09
## 7      sgot 1.40944977692876e-08
## 8        TP    0.398819127523851
## 9       alb 9.07436084295548e-05
## 10 AGratio 8.25114350471533e-05
```

```
ILPD2 = subset(ILPD, select = -c(TP))
ILPD2$liver[ILPD2$liver == 2] = 0
ILPD2$liver = as.integer(ILPD2$liver)
```

3. Variable Selection

(a)

```
formula = liver~age+gender+TB+DB+alkphos+sgpt+sgot+alb+AGratio
fit.full = glm(formula, family = binomial, data = ILPD2);
```

(b)

```
library(MASS)
fit = step(fit.full, direction = "both", k=log(nrows));
```

```
## Start:  AIC=643.08
## liver ~ age + gender + TB + DB + alkphos + sgpt + sgot + alb + 
##     AGratio
## 
##           Df Deviance    AIC
## - alb      1   579.41 636.72
## - TB       1   579.41 636.72
## - gender   1   579.42 636.74
## - AGratio  1   580.42 637.73
## - sgot     1   580.58 637.90
## - DB       1   581.41 638.73
## - alkphos  1   582.64 639.95
## - sgpt     1   584.44 641.75
## <none>         579.39 643.08
## - age      1   588.10 645.41
## 
## Step:  AIC=636.72
## liver ~ age + gender + TB + DB + alkphos + sgpt + sgot + AGratio
## 
##           Df Deviance    AIC
## - TB       1   579.42 630.36
## - gender   1   579.43 630.37
## - sgot     1   580.58 631.53
## - AGratio  1   581.07 632.02
## - DB       1   581.43 632.38
## - alkphos  1   582.68 633.63
## - sgpt     1   584.54 635.49
## <none>         579.41 636.72
## - age      1   588.32 639.27
## + alb      1   579.39 643.08
##
## Step:  AIC=630.36
## liver ~ age + gender + DB + alkphos + sgpt + sgot + AGratio
##
##           Df Deviance    AIC
## - gender   1   579.44 624.02
## - sgot     1   580.59 625.17
## - AGratio  1   581.09 625.66
## - alkphos  1   582.69 627.27
## - sgpt     1   584.55 629.13
## <none>         579.42 630.36
## - age      1   588.34 632.92
## + TB       1   579.41 636.72
## + alb      1   579.41 636.72
## - DB       1   600.65 645.23
##
## Step:  AIC=624.02
## liver ~ age + DB + alkphos + sgpt + sgot + AGratio
##
##           Df Deviance    AIC
## - sgot     1   580.63 618.84
## - AGratio  1   581.09 619.30
## - alkphos  1   582.72 620.93
## - sgpt     1   584.61 622.82
## <none>         579.44 624.02
## - age      1   588.43 626.64
## + gender   1   579.42 630.36
## + TB       1   579.43 630.37
## + alb      1   579.43 630.38
## - DB       1   600.94 639.15
##
## Step:  AIC=618.84
## liver ~ age + DB + alkphos + sgpt + AGratio
##
##           Df Deviance    AIC
## - AGratio  1   582.34 614.18
## - alkphos  1   584.02 615.86
## <none>         580.63 618.84
## - age      1   589.66 621.50
## + sgot     1   579.44 624.02
## + gender   1   580.59 625.17
## + TB       1   580.62 625.20
## + alb      1   580.63 625.21
## - sgpt     1   603.56 635.40
## - DB       1   608.61 640.45
##
## Step:  AIC=614.18
## liver ~ age + DB + alkphos + sgpt
##
##           Df Deviance    AIC
## - alkphos  1   586.96 612.44
## <none>         582.34 614.18
## - age      1   593.10 618.57
## + AGratio  1   580.63 618.84
## + sgot     1   581.09 619.30
## + alb      1   581.50 619.71
## + gender   1   582.32 620.53
## + TB       1   582.33 620.54
## - sgpt     1   604.72 630.19
## - DB       1   612.87 638.34
##
## Step:  AIC=612.44
## liver ~ age + DB + sgpt
##
##           Df Deviance    AIC
## <none>         586.96 612.44
## + alkphos  1   582.34 614.18
## + AGratio  1   584.02 615.86
## - age      1   597.90 617.01
## + sgot     1   585.54 617.38
## + alb      1   585.68 617.53
## + gender   1   586.95 618.79
## + TB       1   586.95 618.80
## - sgpt     1   615.41 634.51
## - DB       1   623.83 642.93
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = liver ~ age + DB + sgpt, family = binomial, data = ILPD2)
##
## 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
```

```
fit.step = glm(liver~age+DB+sgpt, family=binomial, data=ILPD2)
```

(c)

```
library(ncvreg)
y = ILPD2$liver
X = model.matrix(~ age + gender + TB + DB + alkphos + sgpt + sgot + alb + AGratio, data=ILPD2)
colnames(X) = c("(Intercept)", "age", "gender", "TB", "DB", "alkphos", "sgpt", "sgot", "alb",
                "AGratio")
cvfit.SCAD = cv.ncvreg(X=X,y=y, nfolds=5, family="binomial", penalty="SCAD",
lambda.min=.01, nlambda=100, eps=.01, max.iter=1000)
result.SCAD = cvfit.SCAD$fit


beta.hat = as.vector(result.SCAD$beta[-1, cvfit.SCAD$min])
cutoff = 0
terms = colnames(X)[abs(beta.hat) > cutoff]
formula.SCAD = as.formula(paste(c("liver~ 1", terms), collapse=" + "))
fit.pen = glm(formula.SCAD, data = ILPD2, family="binomial")
```

4.

(a)

```
p.jk.full = rep(0, nrows)
p.jk.step = rep(0, nrows)
p.jk.pen = rep(0, nrows)

for (i in 1:nrows){
    fit.jk.full = glm(formula(fit.full), data=ILPD2[-i,], family = "binomial")
    fit.jk.step = glm(formula(fit.step), data=ILPD2[-i,], family = "binomial")
    fit.jk.pen = glm(formula(fit.pen), data=ILPD2[-i,], family = "binomial")

    p.jk.full[i] = predict(fit.jk.full, newdata=ILPD2[i,], type="response")
    p.jk.step[i] = predict(fit.jk.step, newdata=ILPD2[i,], type="response")
  p.jk.pen[i] = predict(fit.jk.pen, newdata=ILPD2[i,], type="response")

}
```

(b)

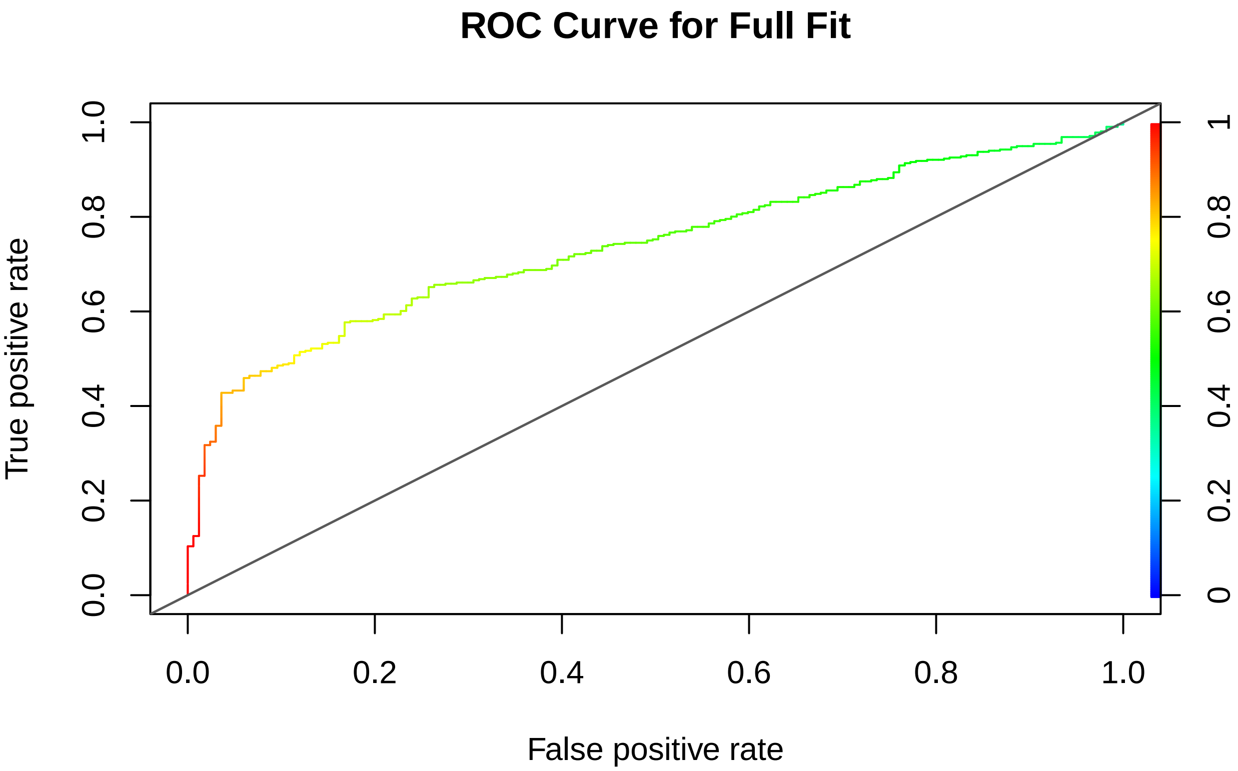
```
library(ROCR)
library(cvAUC)
```

```
##
## Attaching package: 'cvAUC'

## The following object is masked from 'package:ncvreg':
##
##     AUC
```

```
yobs = ILPD2$liver

pred.full = prediction(predictions=p.jk.full, labels=yobs)
perf.full = performance(pred.full,"tpr","fpr")
plot(perf.full, main="ROC Curve for Full Fit", colorize=T)
abline(a=0, b=1, col="gray35", lwd=1.2)
```



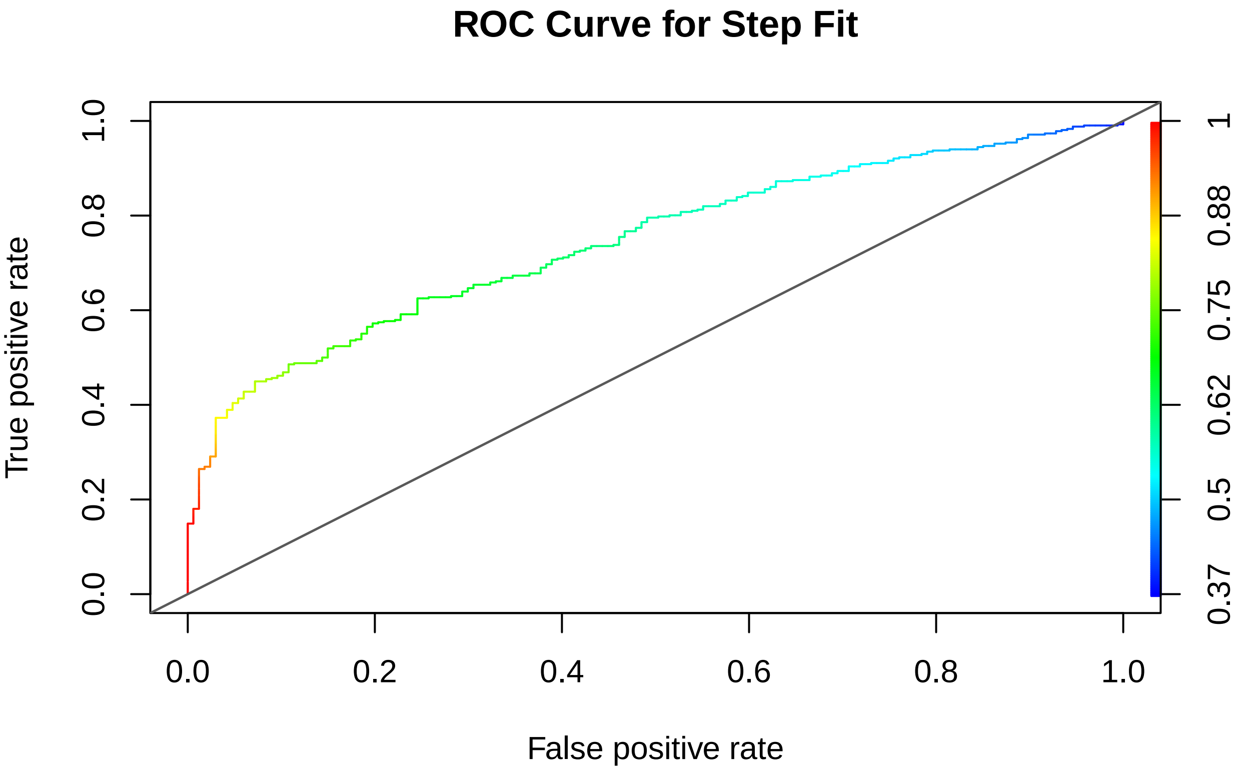
```
AUC = ci.cvAUC(predictions=p.jk.full, labels=yobs, folds=1:nrows, confidence=0.95)
"Full Fit"
```

```
## [1] "Full Fit"
```

```
AUC$cvAUC
```

```
## [1] 0.7376209
```

```
pred.step = prediction(predictions=p.jk.step, labels=yobs)
perf.step = performance(pred.step,"tpr","fpr")
plot(perf.step, main="ROC Curve for Step Fit", colorize=T)
abline(a=0, b=1, col="gray35", lwd=1.2)
```



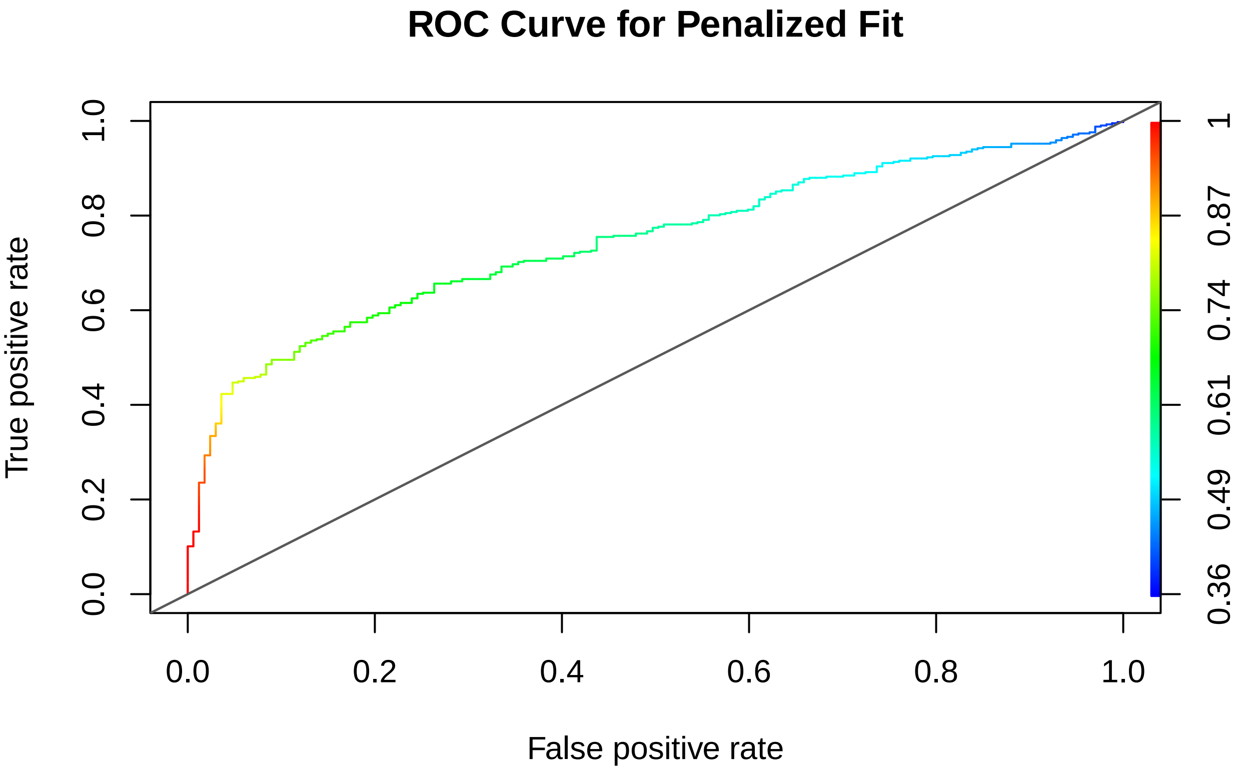
```
AUC = ci.cvAUC(predictions=p.jk.step, labels=yobs, folds=1:nrows, confidence=0.95)
"Step Fit"
```

```
## [1] "Step Fit"
```

```
AUC$cvAUC
```

```
## [1] 0.7439544
```

```
pred.pen = prediction(predictions=p.jk.pen, labels=yobs)
perf.pen = performance(pred.pen,"tpr","fpr")
plot(perf.pen, main="ROC Curve for Penalized Fit", colorize=T)
abline(a=0, b=1, col="gray35", lwd=1.2)
```



```
AUC = ci.cvAUC(predictions=p.jk.pen, labels=yobs, folds=1:nrows, confidence=0.95)
"Penalized Fit"
```

```
## [1] "Penalized Fit"
```

```
AUC$cvAUC
```

```
## [1] 0.7461855
```

5.

```
library(MASS)
fit.pen$coefficients
```

```
## (Intercept)         age  genderMale          DB     alkphos        sgpt
## -0.82019511  0.01848005  0.04587135  0.55985493  0.00130330  0.01344319
##     AGratio
## -0.48022914
```

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

```
## (Intercept)         age  genderMale          DB     alkphos        sgpt
##   0.4403457   1.0186519   1.0469397   1.7504186   1.0013041   1.0135340
##     AGratio
##   0.6186416
```

```
ci = confint(fit.pen, level = 0.95)
```

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

```
ci
```

```
##                      2.5 %      97.5 %
## (Intercept) -1.978945e+00 0.306214011
## age          6.330606e-03 0.030875477
## genderMale  -4.059849e-01 0.491031901
## DB           2.741057e-01 0.943786651
## alkphos     -7.462526e-05 0.003064461
## sgpt         6.626220e-03 0.021700228
## AGratio     -1.190492e+00 0.237610644
```

```
exp(ci)
```

```
##                 2.5 %   97.5 %
## (Intercept) 0.1382150 1.358273
## age         1.0063507 1.031357
## genderMale  0.6663202 1.634001
## DB          1.3153539 2.569694
## alkphos     0.9999254 1.003069
## sgpt        1.0066482 1.021937
## AGratio     0.3040716 1.268215
```

The second output in the cell above gives the odds ratio when a variable is increased by one unit. For example, increasing age by 1 year increases the likelihood of having liver disease by a factor of 1.0187 (if all other variables are equal). This makes sense intuitively because the chances of developing most pathologies increase as we age—for instance, think of heart disease. With this information, it seems that increased age, DB, alkphos, and sgpt all increase the chances that someone has liver disease. The only variable from our model that seems to decrease the chance of having liver disease is AGratio.