STAT645 - Homework 6

Salih Kilicli

10/1/2019

Please obtain the heart disease data (from Course Content > Data). This database from Cleveland clinic (through kaggle) contains 14 attributes. The "target" field refers to the presence of heart disease in the patient. It is an integer, 0 (absent) to 1 (presence). A good description of the attributes can be found here https://lucdemortier.github.io/prxojects/3_mcnulty (https://lucdemortier.github.io/prxojects/3_mcnulty).

Problem 1: There are four categorical $variables, cp, restecg, slope$ **and** $thal$**. Categorize** $thal$ **into two groups, 0 (thal = 3) and 1(thal other than 3).**

```
setwd("/Users/younique/Desktop/STAT645/Data")
heart = read.csv("heart.csv", header=TRUE)
attach(heart)
thall=ifelse(thal==3, 1, 0)
thall
```

```
##   [1] 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0
##  [36] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
##  [71] 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0
## [106] 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0 1 1
## [176] 1 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1
## [211] 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0 1 1 0 0 1 0
## [246] 1 1 0 1 1 1 1 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 1 1 1 0 0
## [281] 0 0 0 1 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0
```

Problem 2: Scale all numeric variables. Do not scale the binary and categorical variables.

```
mydata=data.frame(target=target, sex=sex, fbs=fbs, exang=exang, thal=thall,
  cp=as.factor(cp), slope=as.factor(slope), restecg=as.factor(restecg),
  age=as.vector(scale(age)),
  trestbps=as.vector(scale(trestbps)),
  chol=as.vector(scale(chol)),
  thalach=as.vector(scale(thalach)),
  oldpeak=as.vector(scale(oldpeak)),
  ca=as.vector(scale(ca))
)
```

Problem 3: Fit a logistic regression model to target on 13 **explanatory variables.**

```
logmodel=glm(target~age+sex+cp+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope+ca+thal, data=mydata, family
  ="binomial")
summary(logmodel)
```

```
##
## Call:
## glm(formula = target ~ age + sex + cp + trestbps + chol + fbs +
##     restecg + thalach + exang + oldpeak + slope + ca + thal,
##     family = "binomial", data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7685  -0.3514   0.1533   0.5395   2.6083
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.071109   0.941607   1.138 0.255316
## age         -0.002564   0.214504  -0.012 0.990463
## sex         -1.524191   0.503650  -3.026 0.002476 **
## cp1          1.002568   0.560974   1.787 0.073906 .
## cp2          1.958901   0.474840   4.125 3.70e-05 ***
## cp3          2.045721   0.643097   3.181 0.001467 **
## trestbps    -0.296879   0.186657  -1.591 0.111721
## chol        -0.214911   0.201057  -1.069 0.285112
## fbs          0.092197   0.548060   0.168 0.866406
## restecg1     0.554925   0.373181   1.487 0.137011
## restecg2    -0.271845   2.262329  -0.120 0.904355
## thalach      0.392718   0.244829   1.604 0.108703
## exang       -0.784319   0.424408  -1.848 0.064598 .
## oldpeak     -0.566317   0.261262  -2.168 0.030187 *
## slope1      -0.746528   0.858521  -0.870 0.384545
## slope2       0.188455   0.929521   0.203 0.839335
## ca          -0.838817   0.207814  -4.036 5.43e-05 ***
## thal        -1.350081   0.389862  -3.463 0.000534 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 417.64  on 302  degrees of freedom
## Residual deviance: 202.39  on 285  degrees of freedom
## AIC: 238.39
##
## Number of Fisher Scoring iterations: 6
```

Problem 4: Use this fitted model to estimate the probability of the disease (target= 1) for the following set of values of the explanatory variables. For these cases, also obtain the 95% **interval for the chance of the disease. Note that before the prediction, don't forget to apply the same transformation on the explanatory variables as you have done before the logistic model fitting to the data in the previous question.**

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
68	1	3	145	233	1	0	150	0	2.3	0	0	3
75	0	3	145	150	1	0	150	0	2.3	0	0	1
78	1	0	144	193	1	1	90	0	3.4	1	2	3

```
new=data.frame(
  age=as.vector((c(68,75,78)-mean(heart$age))/sd(heart$age)),
  trestbps=as.vector((c(145,145,144)-mean(heart$trestbps))/sd(heart$trestbps)),
  chol=as.vector((c(233,150,193)-mean(heart$chol))/sd(heart$chol)),
  thalach=as.vector((c(150,150,90)-mean(heart$thalach))/sd(heart$thalach)),
  oldpeak=as.vector((c(2.3,2.3,2.4)-mean(heart$oldpeak))/sd(heart$oldpeak)),
  ca=as.vector(c(0,0,2)-mean(heart$ca))/sd(heart$ca),
  sex=c(1,0,1),
  fbs=c(1,1,1),
  exang=c(0,0,0),
  cp=as.factor(c(3,3,0)),
  restecg=as.factor(c(0,0,1)),
  slope=as.factor(c(0,0,1)),
  thal=ifelse(c(3, 1, 3)==3, 1, 0)
)

myout=predict.glm(logmodel, newdata=new, se.fit=TRUE)

P=1/(1+exp(-as.numeric(myout$fit)))  # p*=exp(hat)/1+exp(hat) or 1/(1+exp(-hat))
cat("Probability estimates for the given data set is: \n", P)
```

```
## Probability estimates for the given data set is:
##  0.537195 0.9666088 0.009695473
```

```
lower=1/(1+exp(-(as.numeric(myout$fit)-1.96*as.numeric(myout$se))))
upper=1/(1+exp(-(as.numeric(myout$fit)+1.96*as.numeric(myout$se)))) # CI=1/(1+exp(-{hat+c(-1,1)1.96se(hat)}))
CI=cbind(lower,upper)
print(CI)
```

```
##            lower      upper
## [1,] 0.128783645 0.90113208
## [2,] 0.704737840 0.99715981
## [3,] 0.001268311 0.07018137
```

Problem 5: Check the adequecy of the model using the Hosmer-Lemeshow test. Clearly write out the hypothesis, test statistic and p-value, and conclusion.

```
library(generalhoslem)
```

```
## Loading required package: reshape
```

```
## Loading required package: MASS
```

```
library(reshape)
library(MASS)
logitgof(heart$target,fitted(logmodel))
```

```
## Warning in logitgof(heart$target, fitted(logmodel)): At least one cell
## in the expected frequencies table is < 1. Chi-square approximation may be
## incorrect.
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  heart$target, fitted(logmodel)
## X-squared = 5.5476, df = 8, p-value = 0.6978
```

For the logistic model, the null and alternative hypotheses given as below:

H_0 : The model fits the data well, H_a : The model is not adequate for the data

Since the p-value (69%) found using Hosmer-Lemeshow test is high, we fail to reject the Null hypothesis, i.e., we do not have sufficient evidence to conclude that the model is not adequate for the data.

Problem 6: Consider the first 100 and the last 100 subjects of the data and fit the logistic regression based on these data only. You don't need to re-scale the data again. Just take the above subset of the data that you have created previously.

```
train=data.frame(
age=as.vector(mydata$age[-c(101:203)]),
trestbps=as.vector(mydata$trestbps[-c(101:203)]),
chol=as.vector(mydata$chol[-c(101:203)]),
thalach=as.vector(mydata$thalach[-c(101:203)]),
oldpeak=as.vector(mydata$oldpeak[-c(101:203)]),
ca=as.vector(mydata$ca[-c(101:203)]),
sex=mydata$sex[-c(101:203)],
fbs=mydata$fbs[-c(101:203)],
exang=mydata$exang[-c(101:203)],
target=mydata$target[-c(101:203)],
cp=as.factor(mydata$cp[-c(101:203)]),
restecg=as.factor(mydata$restecg[-c(101:203)]),
slope=as.factor(mydata$slope[-c(101:203)]),
thal=mydata$thal[-c(101:203)]
)

logmodeltrain=glm(target~age+sex+cp+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope+ca+thal, data=train, fa
  mily="binomial")

summary(logmodeltrain)
```

```
##
## Call:
## glm(formula = target ~ age + sex + cp + trestbps + chol + fbs +
##     restecg + thalach + exang + oldpeak + slope + ca + thal,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2467  -0.2791   0.0240   0.3671   2.5473
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.98329    1.34604   0.731 0.465081
## age            0.04040    0.29844   0.135 0.892313
## sex           -1.32098    0.69114  -1.911 0.055965 .
## cp1            0.66426    0.73144   0.908 0.363797
## cp2            2.68149    0.71369   3.757 0.000172 ***
## cp3            1.12995    0.76141   1.484 0.137802
## trestbps      -0.70105    0.27220  -2.576 0.010008 *
## chol          -0.09391    0.31026  -0.303 0.762135
## fbs            0.32216    0.72300   0.446 0.655893
## restecg1       0.22599    0.51876   0.436 0.663097
## restecg2     -12.86159 1355.94584  -0.009 0.992432
## thalach        0.87858    0.33583   2.616 0.008894 **
## exang         -1.03627    0.55946  -1.852 0.063987 .
## oldpeak       -0.15764    0.41720  -0.378 0.705543
## slope1        -0.98727    1.26449  -0.781 0.434942
## slope2         0.58976    1.41499   0.417 0.676828
## ca            -1.34167    0.33145  -4.048 5.17e-05 ***
## thal          -1.65193    0.56888  -2.904 0.003686 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 277.26  on 199  degrees of freedom
## Residual deviance: 113.25  on 182  degrees of freedom
## AIC: 149.25
##
## Number of Fisher Scoring iterations: 15
```

Problem 7: Next, apply this fitted model to predict the target variable for the remaining set of observations (test data). Show the confusion matrix for prediction when you use 0.5, 0.6 and 0.7 as the cutoff value and use the cutoff to declare a target equal to one if the estimated probability exceeds the cutoff. Comment on the results.

```
library(e1071)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
test=data.frame(
  age=as.vector(mydata$age)[c(101:203)],
  trestbps=as.vector(mydata$trestbps)[c(101:203)],
  chol=as.vector(mydata$chol)[c(101:203)],
  thalach=as.vector(mydata$thalach)[c(101:203)],
  oldpeak=as.vector(mydata$oldpeak)[c(101:203)],
  ca=as.vector(mydata$ca)[c(101:203)],
  sex=mydata$sex[c(101:203)],
  fbs=mydata$fbs[c(101:203)],
  exang=mydata$exang[c(101:203)],
  target=mydata$target[c(101:203)],
  thal=mydata$thal[c(101:203)],
  cp=as.factor(mydata$cp)[c(101:203)],
  restecg=as.factor(mydata$restecg)[c(101:203)],
  slope=as.factor(mydata$slope)[c(101:203)]
)

myout2=predict.glm(logmodeltrain, newdata=test)

P2=1/(1+exp(-as.numeric(myout2[[1]])))                              # p*=exp(hat)/1+exp(hat) or 1/(1+exp(-hat))
cat("Probability estimate for given data set is = ", P2)
```

```
## Probability estimate for given data set is =  0.5323324
```

```
confusionMatrix(data=as.factor(as.numeric(myout2>0.5)), reference=as.factor(test$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28 17
##          1 10 48
##
##                Accuracy : 0.7379
##                  95% CI : (0.642, 0.8196)
##     No Information Rate : 0.6311
##     P-Value [Acc > NIR] : 0.01443
##
##                   Kappa : 0.4578
##
##  Mcnemar's Test P-Value : 0.24821
##
##             Sensitivity : 0.7368
##             Specificity : 0.7385
##          Pos Pred Value : 0.6222
##          Neg Pred Value : 0.8276
##              Prevalence : 0.3689
##          Detection Rate : 0.2718
##    Detection Prevalence : 0.4369
##       Balanced Accuracy : 0.7377
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(data=as.factor(as.numeric(myout2>0.6)), reference=as.factor(test$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29 17
##          1  9 48
##
##                Accuracy : 0.7476
##                  95% CI : (0.6524, 0.828)
##     No Information Rate : 0.6311
##     P-Value [Acc > NIR] : 0.008167
##
##                   Kappa : 0.4806
##
##  Mcnemar's Test P-Value : 0.169811
##
##             Sensitivity : 0.7632
##             Specificity : 0.7385
##          Pos Pred Value : 0.6304
##          Neg Pred Value : 0.8421
##              Prevalence : 0.3689
##          Detection Rate : 0.2816
##    Detection Prevalence : 0.4466
##       Balanced Accuracy : 0.7508
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(data=as.factor(as.numeric(myout2>0.7)), reference=as.factor(test$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29 18
##          1  9 47
##
##                Accuracy : 0.7379
##                  95% CI : (0.642, 0.8196)
##     No Information Rate : 0.6311
##     P-Value [Acc > NIR] : 0.01443
##
##                   Kappa : 0.4634
##
##  Mcnemar's Test P-Value : 0.12366
##
##             Sensitivity : 0.7632
##             Specificity : 0.7231
##          Pos Pred Value : 0.6170
##          Neg Pred Value : 0.8393
##              Prevalence : 0.3689
##          Detection Rate : 0.2816
##    Detection Prevalence : 0.4563
##       Balanced Accuracy : 0.7431
##
##        'Positive' Class : 0
##
```

One measure of goodness of prediction is higher value of Sensitivity+Specifity. Looking at the models with different cutoff values,we see that model with cutoff value 0.6 gives the highest Sensitivity+Specifity value, whereas the model with cutoff value 0.5 has the smallest Sensitivity+Specifity sum. However, all of the models have pretty close Sensitivity and Specifity values.

Problem 8: Draw an ROC curve for the test data mentioned in the previous question and then comment on the discriminatry power of the model.

```
library(MASS)
library(pROC)
```

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

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

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

```
logmodeltest=glm(target ~age+sex+cp+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope+ca+thal, data=test, fam
  ily="binomial")
```

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

```
out=predict(logmodeltest, type="response")
ROC = roc(test$target ~ out)
```

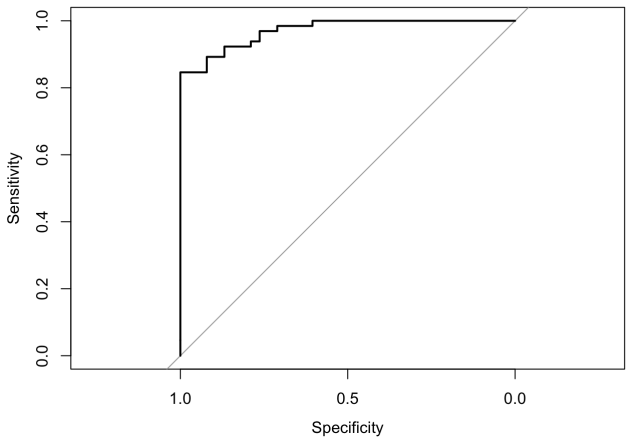
```
## Setting levels: control = 0, case = 1
```

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

```
ROC
```

```
##
## Call:
## roc.formula(formula = test$target ~ out)
##
## Data: out in 38 controls (test$target 0) < 65 cases (test$target 1).
## Area under the curve: 0.9713
```

```
plot(ROC)
```



Looking at the ROC curve, the discriminatry power of the model looks good, close to edges and far from the 45 degrees line.

Problem 9: Re-do the analysis stated in questions 6 and 8 without ca, cp, and $thal$. Comment on the discriminatory power of this model?

```
logmodel2=glm(target~age+sex+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope, data=train, family="binomial"
  )
summary(logmodel2)
```

```
##
## Call:
## glm(formula = target ~ age + sex + trestbps + chol + fbs + restecg +
##     thalach + exang + oldpeak + slope, family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.25970  -0.58795   0.06662   0.64314   2.39701
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.9105     0.9508   3.061 0.002206 **
## age           -0.2264     0.2299  -0.985 0.324741
## sex           -1.6617     0.4952  -3.356 0.000792 ***
## trestbps      -0.6748     0.2244  -3.008 0.002633 **
## chol          -0.1475     0.2049  -0.720 0.471401
## fbs            0.4941     0.5257   0.940 0.347243
## restecg1       0.1523     0.4165   0.366 0.714567
## restecg2     -15.2700  1129.4022  -0.014 0.989213
## thalach        0.8984     0.2657   3.382 0.000720 ***
## exang         -1.4382     0.4204  -3.421 0.000623 ***
## oldpeak       -0.4445     0.2567  -1.731 0.083404 .
## slope1        -1.9277     0.9053  -2.129 0.033234 *
## slope2        -0.9568     0.9936  -0.963 0.335585
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 277.26  on 199  degrees of freedom
## Residual deviance: 167.05  on 187  degrees of freedom
## AIC: 193.05
##
## Number of Fisher Scoring iterations: 15
```

```
myout3=predict.glm(logmodel2, newdata=test)

logmodeltest1=glm(target ~age+sex+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope, data=test, family="binom
  ial")
out=predict(logmodeltest1, type="response")
ROC = roc(test$target ~ out)
```

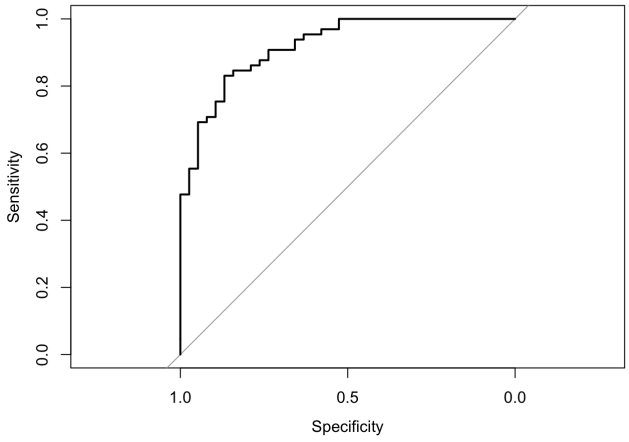
```
## Setting levels: control = 0, case = 1
```

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

```
ROC
```

```
##
## Call:
## roc.formula(formula = test$target ~ out)
##
## Data: out in 38 controls (test$target 0) < 65 cases (test$target 1).
## Area under the curve: 0.9198
```

```
plot(ROC)
```



Looking at the new models ROC curve, it does poorer job than the model compared to the ROC curve since the curve gets closer to 45 degrees line and gets further from the edges. Additionally, clearly the new model has lower Sensitivity values corresponding to lower Specifity values.