Stat 432 Homework 8

Assigned: Oct 11, 2021; Due: 11:59 PM CT, Oct 19, 2021

Contents

Question 1: Logistic Regression

We will use the Cleveland clinic heart disease dataset, which has been used in the lecture note. You can directly download the data from our course website. The goal is to predict the label `num > 0`. The following code prepares the data. I removed `ca` and `thal` because they contain missing values.

```
heart = read.csv("processed_cleveland.csv")
heart$Y = as.factor(heart$num > 0)
heart = subset(heart, select = -c(num, ca, thal))
```

We are going to perform three models:

- A logistic regression
- A logistic regression with Ridge penalty

And we will evaluate them using two different criteria:

- Classification error
- Area under the ROC curve

Also, please note that, to keep things simpler, we will not use cross-validation for this question. Instead, all the evaluations will be just on the training dataset. We are of course at the risk of over-fitting, but Part III will address that issue. In addition, since no cross-validation is needed, you should be using the `glmnet()` function instead of `cv.glmnet()`. The syntax of this function is almost identical to its cross-validation version, except that you will not have the cross-validation feature to help you select the best λ. However, the function will still produce all the coefficients for each λ value. If you need more details, please see the documentation provided at CRAN.

Part I [40 Points]

Complete the following questions for logistic regression:

- Fit logistic regression to the heart data and report the most significant variable.

```r
# Fitting logistic regression to predict Y using all covariates
logistic.fit <- glm(Y ~ ., data = heart, family = binomial)
summary(logistic.fit)
```

```
##
## Call:
## glm(formula = Y ~ ., family = binomial, data = heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3192  -0.6484  -0.2130   0.5886   2.6819
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.191858   2.510711  -2.864  0.00418 **
## age          0.023549   0.020797   1.132  0.25749
## sex          1.955728   0.400915   4.878 1.07e-06 ***
## cp           0.810770   0.179562   4.515 6.32e-06 ***
## trestbps     0.019365   0.009724   1.992  0.04643 *
## chol         0.005369   0.003263   1.646  0.09982 .
## fbs         -0.215480   0.445551  -0.484  0.62865
## restecg      0.201612   0.163232   1.235  0.21678
## thalach     -0.023967   0.009166  -2.615  0.00893 **
## exang        0.989378   0.362936   2.726  0.00641 **
## oldpeak      0.496534   0.183807   2.701  0.00691 **
## slope        0.320861   0.320731   1.000  0.31711
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 417.98  on 302  degrees of freedom
## Residual deviance: 252.43  on 291  degrees of freedom
## AIC: 276.43
##
## Number of Fisher Scoring iterations: 5
```

The highly significant variables are "cp" and "sex" (with high significance codes) and the most significant variable is "sex" as it has the smallest p-value.

- Using 0.5 as the cut-off value of predicted probability, produce the confusion table of the training data. What is the classification error associated with this model?

```r
# Using 0.5 as the cut-off value of predicted probability
yhat = (logistic.fit$fitted.values > 0.5)
# Producing confusion table of the training data
confusion_table = table(yhat, heart$Y)
confusion_table
```

```
##
## yhat    FALSE TRUE
##   FALSE   138   34
##   TRUE     26  105
```

The above is the confusion table using 0.5 as the cut-off value of predicted probability.

```
# Calculating classification error
classification_error = (confusion_table[1, 2] + confusion_table[2, 1]) /nrow(heart)
classification_error
```

```
## [1] 0.1980198
```

Associated classification error with the model is 0.1980

- What is the sensitivity and specificity of this model? Choose a new cut-off value that would give a higher sensitivity, and report the confusion table and sensitivity associated with this new cut-off value.

```
# Calculating sensitivity
sensitivity = confusion_table[2, 2] / (confusion_table[2, 2] + confusion_table[1, 2])
sensitivity
```

```
## [1] 0.7553957
```

```
# Calculating specificity
specificity = confusion_table[1, 1] / (confusion_table[1, 1] + confusion_table[2, 1])
specificity
```

```
## [1] 0.8414634
```

The sensitivity and specificity of this model are 0.7553957 and 0.8414634 respectively.

```
# Using 0.11 as the cut-off value of predicted probability
pred = predict(logistic.fit, newdata = heart, type = "response")
# Producing confusion table of the training data
confusion_table2 = table(pred > 0.11, heart$Y)
confusion_table2
```

```
##
##         FALSE TRUE
##   FALSE    60    3
##   TRUE    104  136
```

The above is the confusion table using 0.11 as the cut-off value of predicted probability.

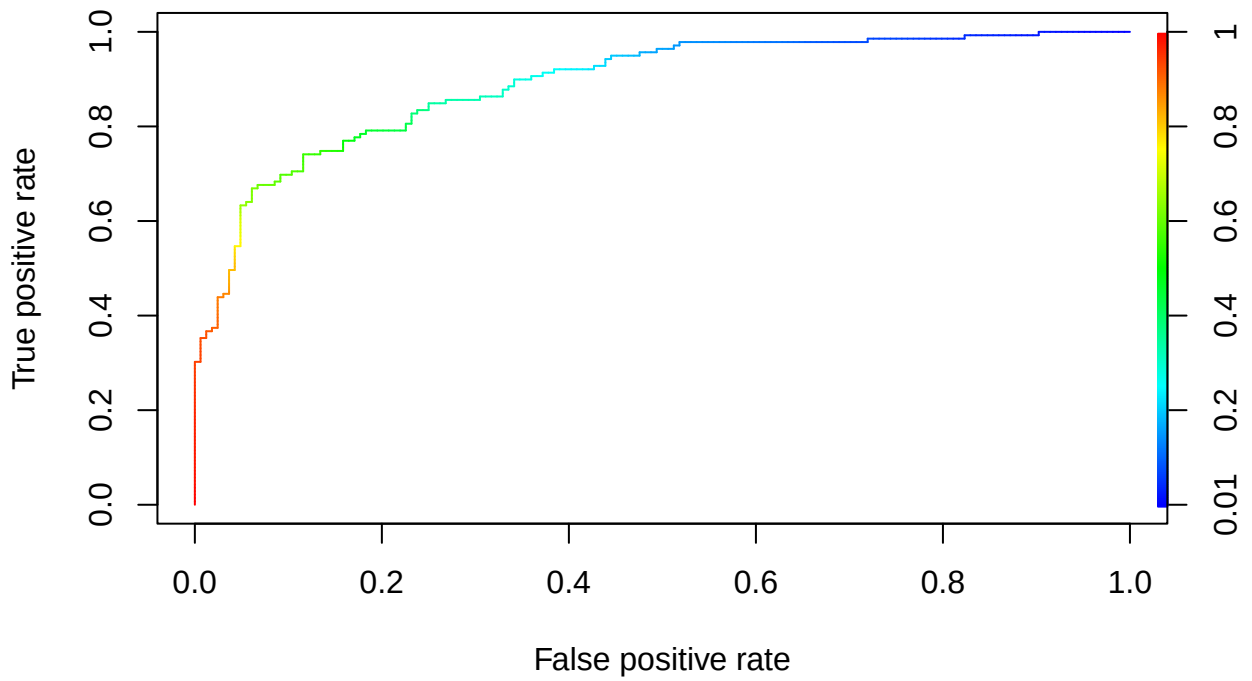
```
# Calculating new sensitivity value
sensitivity2 = confusion_table2[2, 2] / (confusion_table2[2, 2] + confusion_table2[1, 2])
sensitivity2
```

```
## [1] 0.9784173
```

The sensitivity of this model with the cut-off value of 0.11 is 0.9784173 which is much higher than 0.7553957 (previously calculated sensitivity with the cut-off value of 0.5)

- Produce the ROC curve plot associated with your logistic regression and report the AUC.

```
library(ROCR)
roc <- prediction(pred, heart$Y)

# Calculating the ROC curve
perf <- performance(roc, "tpr", "fpr")
plot(perf, colorize = TRUE)
```




```
# Computing the area under the curve
performance(roc, measure = "auc")@y.values[[1]]
```

```
## [1] 0.8893666
```

AUC (area under the curve) is 0.8893666.

Part II [40 Points]

Complete the following questions for logistic regression with Ridge penalty :

- Use the `glmnet()` function to produce a set of coefficients across many λ values.
- Since we will not perform cross-validation, let's just use one of the λ values. You can extract all the coefficients using the `coef()` function. This will give you a matrix of 100 columns, associated with 100 different λ values. Let's use the coefficients associated with the 40th smallest λ value. Based on these coefficients, calculate the predicted (using training data) probabilities of all observations. Use a histogram to plot all of them.
- Using 0.5 as the cut-off value of predicted probability, produce the confusion table of the training data. What is the classification error associated with this model?
- Produce the ROC curve plot associated with your model and report the AUC.

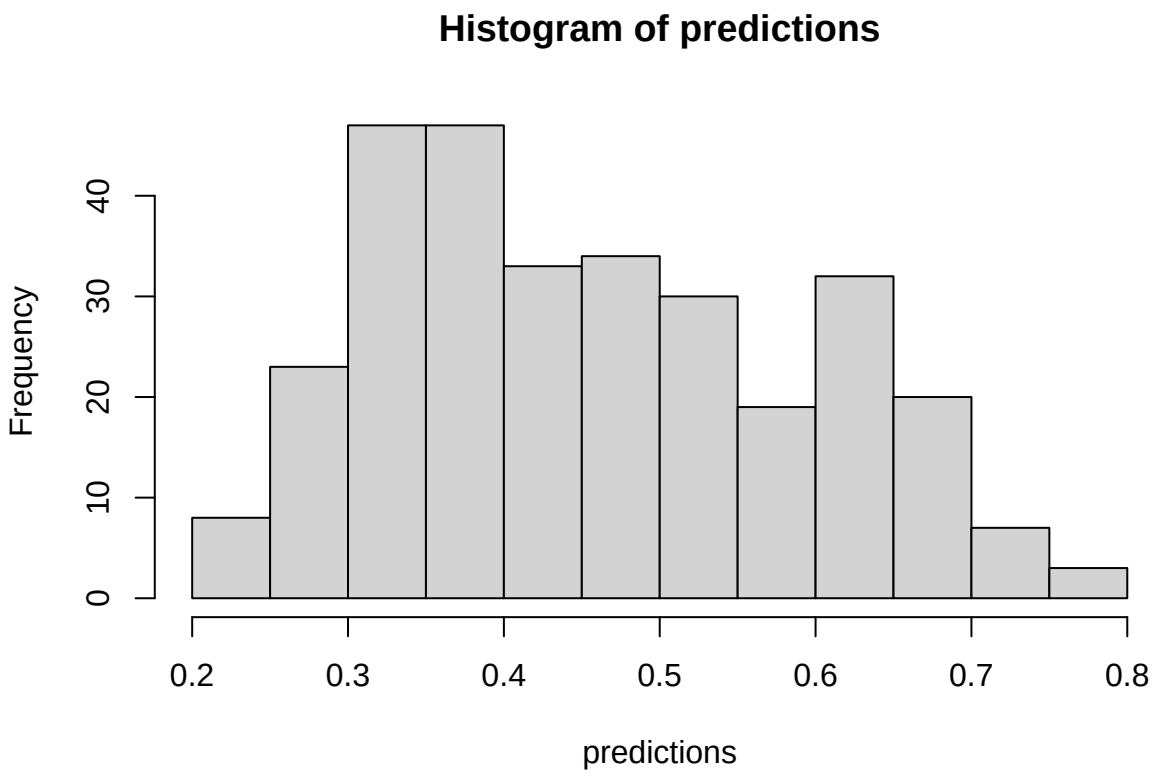
```r
# Loading required library
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```r
# Setting UIN as seed
set.seed(667346304)

# Fitting glmnet() to produce a set of coefficients across many lambda values
fit2 = glmnet(
  x = heart[, 1:11],
  y = heart[, 12],
  alpha = 0,                  # Ridge penalty
  family = "binomial"         # Logistic regression
)
```

```r
# 40th smallest lambda value
lambda_40 = sort(fit2$lambda)[40]
# coefficients associated with the 40th smallest lambda value
coef_40 = coef(fit2, s = lambda_40)

# prepare the data matrix by adding a column of 1 for intercept
x = as.matrix(cbind("intercept" = 1, heart[, 1:11]))
coef_40 = as.matrix(coef_40)

# calculate the predicted probabilities of all observations (using training data)
predictions = exp(x %*% coef_40) / (1 + exp(x %*% coef_40))

# Use a histogram to plot all of them
hist(predictions)
```



The above plot is a histogram of predicted (using training data) probabilities of all observations using the coefficients associated with the 40th smallest lambda value.

```
# Using 0.5 as the cut-off value of predicted probability and produce the confusion table
confusion_table3 <- table(predictions > 0.5, heart$Y)
confusion_table3
```

```
##
##         FALSE TRUE
##   FALSE   149   43
##   TRUE     15   96
```

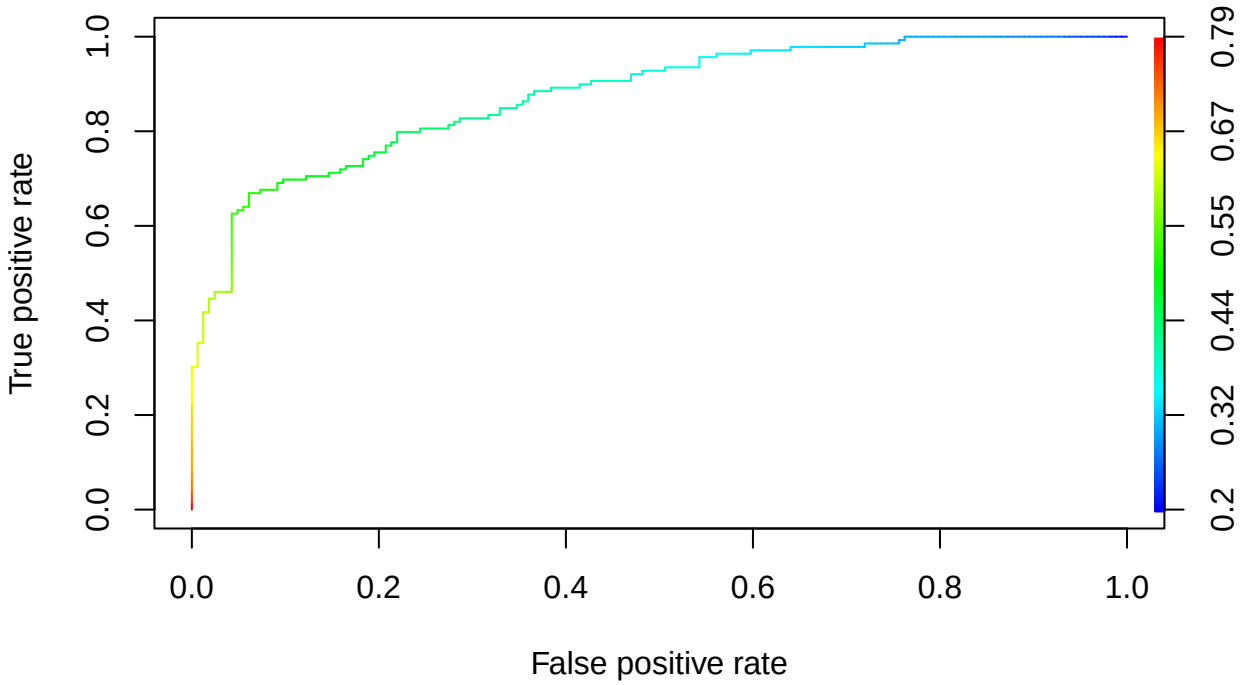
The above is the confusion table using 0.5 as the cut-off value of predicted probability.

```
# classification error associated with this model
classification_error3 = (confusion_table3[1, 2] + confusion_table3[2, 1]) /nrow(heart)
classification_error3
```

```
## [1] 0.1914191
```

The classification error associated with this model is 0.1914191.

```
# Produce the ROC curve plot associated with your model
library(ROCR)
roc2 <- prediction(predictions, heart$Y)
# calculates the ROC curve
perf2 <- performance(roc2, "tpr", "fpr")
plot(perf2, colorize = TRUE)
```



The above is the ROC curve plot associated with the model.

```
# Computing the area under the curve
performance(roc2, measure = "auc")@y.values[[1]]
```

```
## [1] 0.8758993
```

The AUC(area under the curve) is 0.8758993.

Part III [10 Points]

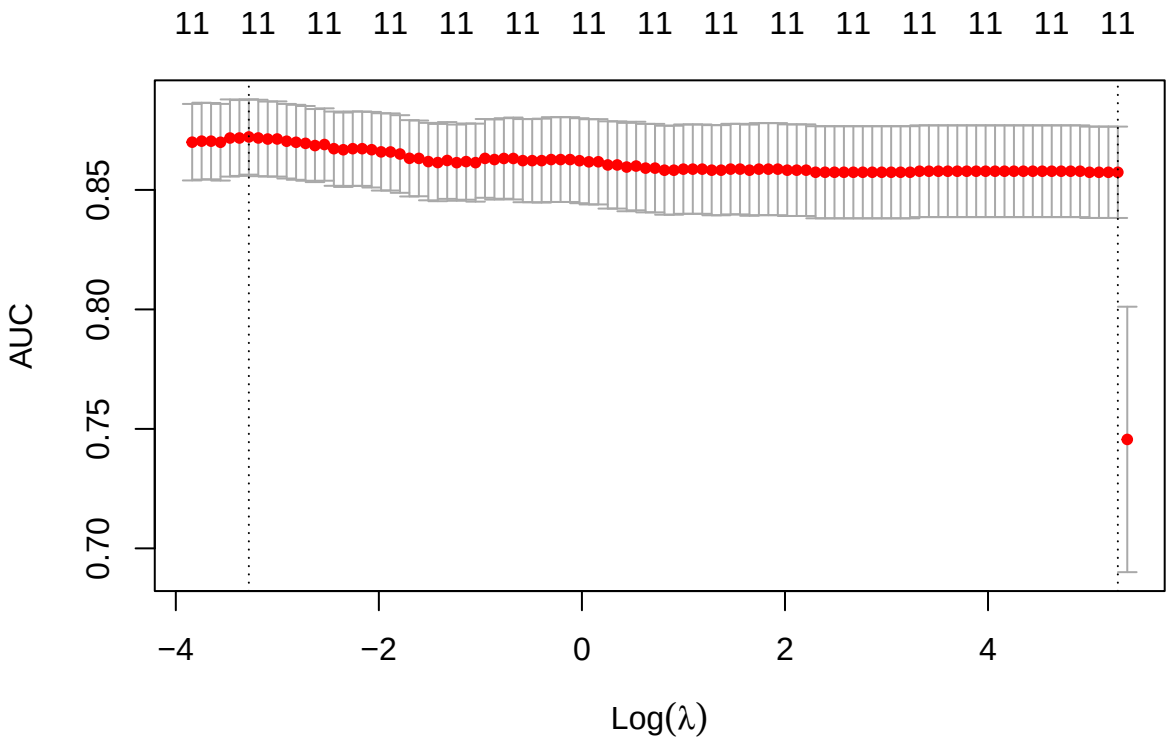
In this last part, we will use a built-in feature of the `glmnet` package. Read the documentation of the `cv.glmnet()` function at CRAN and understand how to specify the `type.measure` argument so that the cross-validation uses the AUC as the selection criterion of λ to pick the best model. Implement a 10-fold cross-validation Ridge regression using our data and report the best λ value (`"lambda.min"`). What is the cross-validation AUC associated with this penalty?

```
# Implementing a 10-fold cross-validation Ridge regression with `type.measure`as AUC
set.seed(667346304)
fit3=cv.glmnet(x=as.matrix(heart[, 1:11]),
          y=as.matrix(heart[,12]),
          type.measure = "auc",          # `type.measure`as AUC
          alpha = 0,                     # alpha=0 : Ridge
          nfolds = 10,                   # 10-fold cv
          family="binomial")

# Report the best lambda value (`"lambda.min"`)
fit3$lambda.min
```

```
## [1] 0.03760868
```

The best lambda value ("lambda.min") with cross-validation using the AUC is 0.03760868.

```
plot(fit3)
```




```
# Reporting cross-validation AUC associated with best penalty
fit3$cvm[which(fit3$lambda == fit3$lambda.min)]
```

```
## [1] 0.8721693
```

The cross-validation AUC associated with the best λ value (`"lambda.min"`) is 0.8721693.