Classification Lab

Kristine Dinh

2020-10-29

## Task 1: EDA; response is ‘default’

Summarize the response, the relationship between the response and the three explanatory variables, and the relationship between the three explanatory variables.

### Code set-up

Here is a code chunk with suggested summary statistics.

table(default) # 97% did not default

## default  
## No Yes   
## 9667 333

The target variable default has high proportion of not default (97%) and low proportion of default (3%). Here, the proportion of default and non-default are not balance. This could cause the model to be unreliable.

# Numeric variables: balance, income  
# I have coded this table in completion to present desired structure and labeling.  
# Do something similar for the breakdown of balance and of income by default status.  
balance\_s = c(median(balance), mean(balance), sd(balance))  
income\_s = c(median(income), mean(income), sd(income))  
numeric\_table = rbind(balance\_s, income\_s)  
colnames(numeric\_table) = c("Mean", "Median", "Std dev")  
rownames(numeric\_table) = c("Balance", "Income")  
print(numeric\_table)

## Mean Median Std dev  
## Balance 823.637 835.3749 483.715  
## Income 34552.645 33516.9819 13336.640

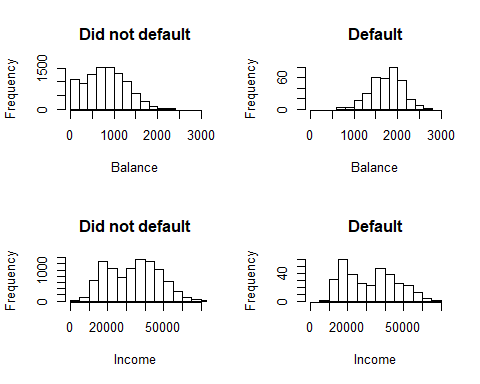
The mean and median values of balance and income are 823.64 and 3.45526410^{4}. Standard deviation for both continuous variable are quite large suggesting the ranges between each observation are large.

## by default status  
numerics\_default = cbind(  
 paste(signif(tapply(balance, default, mean, na.rm=T), digits=3), "(", signif(tapply(balance, default, sd, na.rm=T), digits=3), ")"),  
 paste(signif(tapply(income, default, mean, na.rm=T), digits=3), "(", signif(tapply(income, default, sd, na.rm=T), digits=3), ")"))  
rownames(numerics\_default) = c("Did not default", "Defaulted")  
colnames(numerics\_default) = c("Balance", "Income")  
print(numerics\_default)

## Balance Income   
## Did not default "804 ( 456 )" "33600 ( 13300 )"  
## Defaulted "1750 ( 341 )" "32100 ( 13800 )"

The mean for balance is higher when there is a “Yes” in default and lower otherwise. This suggest a postive relationship between balance and default. Meaning when balance is higher, default will more likely to be “Yes”. On the other hand, the mean of income is higher when there is a “No” in default and lower otherwise suggesting a negative relationship. If income increases, default will more likely to be “No”.

par(mfrow=c(2,2))  
hist(balance[Default=="No"], xlim=c(0,3000), main="Did not default", xlab="Balance")  
hist(balance[Default=="Yes"], xlim=c(0,3000), xlab="Balance", main="Default")  
hist(income[Default=="No"], xlim=c(0, 70000), xlab="Income", main="Did not default")  
hist(income[Default=="Yes"], xlim=c(0,70000), xlab="Income", main="Default")



As looking at the histograms of default factors by balance, we can see that the mode of balance smaller when there is default is “No” and higher when default is “Yes”. This suggests that lower balance leads to a “No” in default, and higher balance leads to a “Yes” in default. Looking at the plot of income by default, as default tends to be “No” as income increases and “Yes” as income decreases. This shows that lower income would lead to a “Yes” in default and higher income would lead to a “No”.

# Categorical variable: student  
table(student)

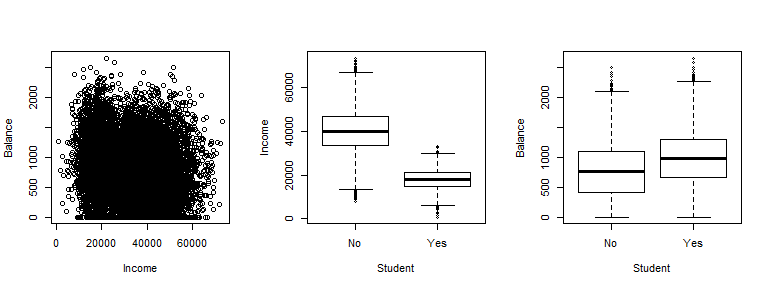
## student  
## No Yes   
## 7056 2944

table(student, default)

## default  
## student No Yes  
## No 6850 206  
## Yes 2817 127

In the contingency table of student and default, we can see that a student would be less likely to have a default. Similarly, a non-student would have more chances of getting a “No” in default. There are more non-student (7056) than student (2944).

# Relationships between explanatory variables  
par(mfrow=c(1,3))  
plot(income, balance, ylab="Balance", xlab="Income")  
boxplot(income~student, ylab="Income", xlab="Student")  
boxplot(balance~student, ylab="Balance", xlab="Student")



Income and balance scatter plot seems to have an interaction effect. This mean that higher income can have both high balance and low balance. Both variables income and balance seems to have low correlation with each other. Box plots shows students have lower income and higher balance. The mean of income between student vs non-student seems to have a large gap suggesting a high correlation.

# Correlations  
allPairs = rbind(t(combn(2:ncol(Default), 2)), matrix(data = c(2:ncol(Default),2:ncol(Default)), ncol=2))  
allPairs = allPairs[order(allPairs[,1], allPairs[,2]),]  
rhos = apply(X=allPairs, 1, FUN = function(X, Y) cor.test(as.numeric(Y[,X[1]]),as.numeric(Y[,X[2]]), method = "spearman", exact=F)$estimate, Y=Default)  
matP.Both = matrix(nrow = ncol(Default)-1, ncol = ncol(Default)-1)  
#Create matrix to store p values  
matP.Both[allPairs-1] = rhos  
colnames(matP.Both) = names(Default)[2:ncol(Default)]  
rownames(matP.Both) = names(Default)[2:ncol(Default)]  
signif(matP.Both, digits=3)

## student balance income  
## student 1 0.198 -0.753  
## balance NA 1.000 -0.149  
## income NA NA 1.000

Correlation matrix shows there is a strong negative correlation between income and student. This means there might be multicolinearity issues if we have these two variables in the same models.

## Task 2, model comparison

Consider two models:

* Model 1, logistic regression model of default on income and balance
* Model 2, logistic regression model of default on income, balance, and student status.

Perform training/testing evaluation of Models 1 and 2. Suggested measures:

* Confusion matrix
* ROC curve
* Sensitivity and specificity at an “optimal” cutoff from ROC curve

### Code set-up

The following two code chunks provide, first, measures for Model 1 and then second a chunk presenting the desired output. Replicate the first code chunk for Model 2 and then add the appropriate measures into the ROC\_output code chunk for presentation. Make sure all R outputted tables have well-labeled columns and rows, use colnames and rownames functions.

n = dim(Default)[1] # sample size  
  
# Split data into training and testing sets  
p = 0.5  
set.seed(1) # set the random number generator seed  
train = sample(n, p\*n) # random sample percentage out of n; this creates the index list  
fit\_train1 = glm(default~balance+income, family=binomial(link=logit), data=Default, subset = train)  
test\_probs1 = predict.glm(fit\_train1, Default, type="response")[-train]  
  
fit\_train2 = glm(default~balance+income+student,   
 family=binomial(link=logit), data=Default, subset = train)  
test\_probs2 = predict.glm(fit\_train2, Default, type="response")[-train]  
  
# ROC curve  
# Plot function of ISLR  
rocplot=function(pred, truth, ...){  
 predob = prediction (pred, truth)  
 perf = performance (predob , "tpr", "fpr")   
 plot(perf ,...)}  
  
# Statistics off the ROC  
pred1 = prediction(test\_probs1, Default[-train,"default"])  
pred2 = prediction(test\_probs2, Default[-train,"default"])  
# calculating AUC  
auc1 <- performance(pred1,"auc")  
auc2 <- performance(pred2,"auc")  
  
# convert S4 class to vector  
auc1 <- unlist(slot(auc1, "y.values"))  
auc2 <- unlist(slot(auc2, "y.values"))  
  
# Compute optimal cutoff  
opt.cut = function(perf, pred){  
 cut.ind = mapply(FUN=function(x, y, p){  
 d = (x - 0)^2 + (y-1)^2  
 ind = which(d == min(d))  
 c(sensitivity = y[[ind]], specificity = 1-x[[ind]],   
 cutoff = p[[ind]])  
 }, perf@x.values, perf@y.values, pred@cutoffs)  
}  
# Present sensitivity and specificity for that optimal cutoff  
roc.perf1 = performance(pred1, measure="tpr", x.measure="fpr")  
roc.perf2 = performance(pred2, measure="tpr", x.measure="fpr")  
  
# Here is a function and code that will compute sensitivity and specificity at any given cutoff  
se.sp <- function (cutoff, pred){  
 sens <- performance(pred,"sens")  
 spec <- performance(pred,"spec")  
 num.cutoff <- which.min(abs(sens@x.values[[1]] - cutoff))  
 return(list(Cutoff=sens@x.values[[1]][num.cutoff],  
 Sensitivity=sens@y.values[[1]][num.cutoff],  
 Specificity=spec@y.values[[1]][num.cutoff]))  
}  
se.sp(.5, pred1) # Sensitivity and specificity at 0.5 cutoff

## $Cutoff  
## 9762   
## 0.504902   
##   
## $Sensitivity  
## [1] 0.3121019  
##   
## $Specificity  
## [1] 0.9960768

se.sp(.5, pred2)

## $Cutoff  
## 5242   
## 0.490623   
##   
## $Sensitivity  
## [1] 0.2866242  
##   
## $Specificity  
## [1] 0.9960768

# Confusion matrix at 0.5 cutoff  
c = 0.5  
test\_class1 = (test\_probs1 > c)  
table(test\_class1, Default$default[-train], dnn=c("Predicted", "Model 1 Truth")) # cross-classification accuracy

## Model 1 Truth  
## Predicted No Yes  
## FALSE 4824 108  
## TRUE 19 49

paste("Accuracy of Model 1 is", sum(as.numeric(test\_class1) == (as.numeric(Default$default[-train])-1))/(n-n\*p))

## [1] "Accuracy of Model 1 is 0.9746"

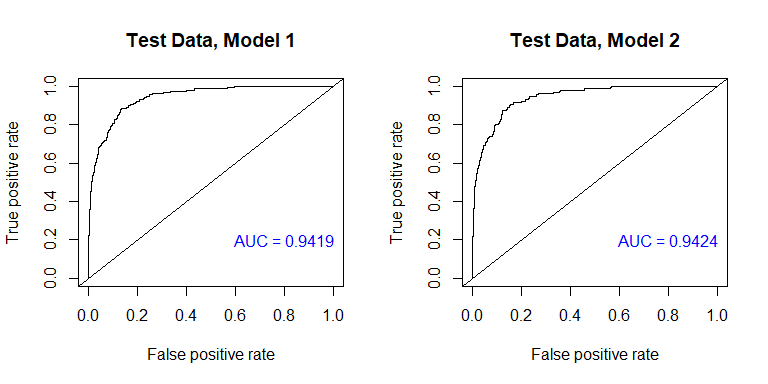
## [Confusion matrix code for Model 2 here]  
  
test\_class2 = (test\_probs2 > c)  
table(test\_class2, Default$default[-train], dnn=c("Predicted", "Model 2 Truth")) # cross-classification accuracy

## Model 2 Truth  
## Predicted No Yes  
## FALSE 4825 112  
## TRUE 18 45

paste("Accuracy of Model 2 is", sum(as.numeric(test\_class2) == (as.numeric(Default$default[-train])-1))/(n-n\*p))

## [1] "Accuracy of Model 2 is 0.974"

# ROC curves  
par(mfrow=c(1,2))  
rocplot(test\_probs1, Default[-train,"default"], main="Test Data, Model 1")  
abline(a=0, b=1)  
text(0.8, 0.2, paste("AUC =", signif(auc1, digits=4)), col="blue")  
## [ROC plot code for Model 2 here]  
rocplot(test\_probs2, Default[-train,"default"], main="Test Data, Model 2")  
abline(a=0, b=1)  
text(0.8, 0.2, paste("AUC =", signif(auc2, digits=4)), col="blue")



# Sensitivity and specificity at an optimal cutoff  
print("Model 1: Sensitivities and specificities at the optimal cutoff:")

## [1] "Model 1: Sensitivities and specificities at the optimal cutoff:"

signif(opt.cut(roc.perf1, pred1), digits=2)

## [,1]  
## sensitivity 0.890  
## specificity 0.860  
## cutoff 0.035

## [Add code to present this in a table with the analogous values for Model 2]  
print("Model 2: Sensitivities and specificities at the optimal cutoff:")

## [1] "Model 2: Sensitivities and specificities at the optimal cutoff:"

signif(opt.cut(roc.perf2, pred2), digits=2)

## [,1]  
## sensitivity 0.870  
## specificity 0.880  
## cutoff 0.041

## Task 3, K-fold cross-validation evaluation of the two models

Recommend 10-fold cross-validation, as leave-one-out is slow on this size data set.

### Code set-up

This code chunk performs 10-fold cross validation error analysis for model 1. Perform an analogous analysis for Model 2.

# First, 10-fold cv on Model 1 (default on balance and income)  
fit1 = glm(default~balance+income, family=binomial(link=logit), data=Default)  
summary(fit1)

##   
## Call:  
## glm(formula = default ~ balance + income, family = binomial(link = logit),   
## data = Default)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4725 -0.1444 -0.0574 -0.0211 3.7245   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.154e+01 4.348e-01 -26.545 < 2e-16 \*\*\*  
## balance 5.647e-03 2.274e-04 24.836 < 2e-16 \*\*\*  
## income 2.081e-05 4.985e-06 4.174 2.99e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2920.6 on 9999 degrees of freedom  
## Residual deviance: 1579.0 on 9997 degrees of freedom  
## AIC: 1585  
##   
## Number of Fisher Scoring iterations: 8

set.seed(17) # set the random number generator seed  
# 10-fold cv, compute misclassification rate  
# Note that this R function also provides a second component with a bias adjustment  
cv.error.10 = cv.glm(Default, fit1, K=10)   
paste("The cv error for Model 1 is", signif(cv.error.10$delta[1], digits=3))

## [1] "The cv error for Model 1 is 0.0215"

# First, 10-fold cv on Model 2 (default on balance and income)  
fit2 = glm(default~balance+income+student, family=binomial(link=logit), data=Default)  
summary(fit2)

##   
## Call:  
## glm(formula = default ~ balance + income + student, family = binomial(link = logit),   
## data = Default)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4691 -0.1418 -0.0557 -0.0203 3.7383   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.087e+01 4.923e-01 -22.080 < 2e-16 \*\*\*  
## balance 5.737e-03 2.319e-04 24.738 < 2e-16 \*\*\*  
## income 3.033e-06 8.203e-06 0.370 0.71152   
## studentYes -6.468e-01 2.363e-01 -2.738 0.00619 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2920.6 on 9999 degrees of freedom  
## Residual deviance: 1571.5 on 9996 degrees of freedom  
## AIC: 1579.5  
##   
## Number of Fisher Scoring iterations: 8

set.seed(17) # set the random number generator seed  
# 10-fold cv, compute misclassification rate  
# Note that this R function also provides a second component with a bias adjustment  
cv.error.10 = cv.glm(Default, fit2, K=10)   
paste("The cv error for Model 2 is", signif(cv.error.10$delta[1], digits=3))

## [1] "The cv error for Model 2 is 0.0214"

### Report the following for Tasks 2 and 3:

Write text to compare the two models with respect to ROC curves, corresponding ROC curve statistics (accuracy, sensitivity, and specificity), cross validation error, and regression model fit/inferences. Which model would you choose to predict default status? Why?

Model 1 has an AUC of 0.9419 and model 2 has an AUC of 0.9424 indicating model 2 is a little better. Both models have high amount of AUC meaning the predictions are closed to the empirical values. Sensitivity for model 1 (.89) is higher than model 2 (.87). On the other hand, specificity for model 2 (.88) is better than model model 1 (.86). Meaning model 1 has higher percentage of true defaulters correctly identified and model 2 has higher percentage of non-defaulters correctly identified.

After running cross validation, 2 models have similar cross validation error with 0.001 of differences in error. The differences are not noticible, so both models are similar based on cross validation error.

Model 1 AIC (1585) is slightly higher than model 2 (1579) which made model 2 a better candidate. However, coefficient of income in model 2 is not sigificant. This indicates that income is not needed in the model when balance and student are already in the model.

Therefore, I would choose model 1 to predict default status. Athough model 1 AUC is lower than model 2 and AIC is higher than model 2, the differences are not too high. With two predictors (balance and income) in the model, the predictions can be nearly as good as model 2.

## Task 4: Risk score

Write text to discuss how you would compute a ‘default risk score’ from the model you chose.

Default risk score can be calculated using model 1. We can have a new test set and plug the test set into the predict function with type is “response” to get the proportion of default. Or we can calculate by hand using the coefficient from the summary of regression and plus the new number of income and balance to get the default risk score.