Classification Lab

[YOUR NAME HERE]

## 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.

attach(Default)  
# Summary stats  
# Response, Default: what percentage defaulted?  
table(default) # 97% did not default

## default  
## No Yes   
## 9667 333

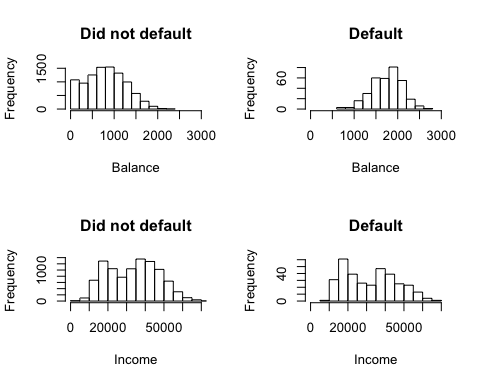
# Numeric variables: balance, income  
# I have coded this table in completion to present desired structure and labeling.  
# Do something similar for the breakdown of balance and of income by default status.  
balance\_s = c(median(balance), mean(balance), sd(balance))  
income\_s = c(median(income), 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

## 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 )"

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")



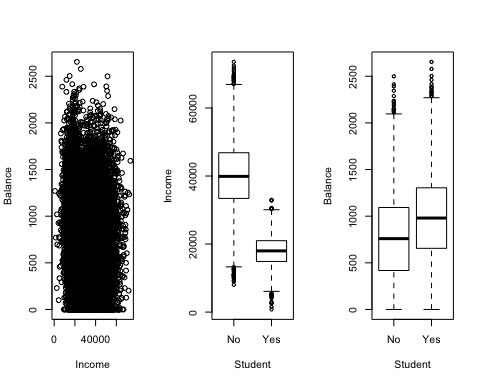
# Categorical variable: student  
table(student)

## student  
## No Yes   
## 7056 2944

table(student, default)

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

# 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")



# 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

### Report the following

Present an EDA description of the data with respect to predicting default from income, balance, and student status. Remember that you can read R output into your text description; for example,

Some summary statistics for credit card balance in this data set is 824, 835, 484.

## 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]  
  
# 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"])  
# calculating AUC  
auc1 <- performance(pred1,"auc")  
# convert S4 class to vector  
auc1 <- unlist(slot(auc1, "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")  
  
# 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

# 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 4805 115  
## TRUE 28 52

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 0.9714"

## [Confusion matrix code for Model 2 here]  
  
# 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]  
  
# Sensitivity and specificity at an optimal cutoff  
print("Sensitivities and specificities at the optimal cutoff:")

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

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

## [,1]  
## sensitivity 0.890  
## specificity 0.870  
## cutoff 0.035

## [Add code to present this in a table with the analogous values for Model 2]



## 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 0.0214"

### 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?

## Task 4: Risk score

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