Assignment 3

Ethan Tenison

3/20/2020

# Q1: Samples and Resampling (40 points)

In Question 2 on Assignment 2, we used the validation set approach to model assessment and selection.

### (a) How many observations were in the sample? In the training set? In the test set? How many

### training sets were there? How many test sets? What are strengths and weaknesses of this

### approach?

There were 4,422 observations in the dataset from assignment 2, and I used two test/training sets. There were 3316 obs in the training and 1106 in the test in the firest with a 75% split. FOr the KNN classifier 3538 and 884 using a 80% 20% split.

The validation approach is simple and easy, but your estimates can have a lot of variation depending on what is inside your training/test set.

###Suppose we had used LOOCV instead of the validation set approach. ###(b) How many observations would there have been in the training set? In the test set? How many ###training sets would there have been? How many test sets? What are strengths and weaknesses ###of this approach?

With LOOCV there would be 4421 observations in training and 1 in test, but there would be 4422 train/test sets. The main advantage is that you will have less bias because n-1 observations are in your training set, but it is computational expensives. You wouldn’t want to use this if you have millions of observations.

### Now suppose we had used k-fold cross-validation instead of the validation set approach.

### (c) How many observations would there have been in the training set? In the test set? How many

### training sets would there have been? How many test sets? What are strengths and weaknesses

### of this approach?

It all depends on how many folds you select. 5 K-fold validation would split the data into 5 random sets of approximately the same size. The biggest appeal for kfold is its speed. It’s less computationally intensive, but there is more bias. Increasing the number of folds reduces bias close to LOOCV.

###Now suppose we had used bootstrapping instead of the validation set approach. ###(d) How many observations would there have been in the training set? In the test set? How many ###training sets would there have been? How many test sets? What are strengths and weaknesses ###of this approach?

The number of observations is randomly assigned in the bootstrap method. The total number of bootstrap samples depends on how much variance is in your statistics. If there is high variance, you might want more samples than if the variance was low.

# Q2: Resampling in data (40 points)

library(reshape2)  
library(readr)  
library(janitor)  
library(dplyr)  
  
water <- read\_csv("Austin\_Water\_-\_Residential\_Water\_Consumption.csv")  
water <- clean\_names(water)  
water$customer\_class <- gsub(" - ", "\_", water$customer\_class)  
water$customer\_class <- gsub("-", "\_", water$customer\_class)  
water$year <- as.numeric(substr(water$year\_month, 1, 4))  
water$month <- as.numeric(substr(water$year\_month, 5, 6))  
water <- water[water$postal\_code != "", ]  
water <- na.omit(water)  
water <- as.data.frame(aggregate(data=water, total\_gallons~customer\_class+year+postal\_code, sum))  
water <- water[water$year == 2014,]  
water$hi\_use <- 0  
water$hi\_use[water$total\_gallons > mean(water$total\_gallons)] = 1  
water$zip\_code <- water$postal\_code  
  
houses <- read\_csv("2014\_Housing\_Market\_Analysis\_Data\_by\_Zip\_Code.csv")  
houses <- clean\_names(houses)  
atx <- merge(water, houses, by="zip\_code", all.x=TRUE)

###(a) Split the data into a training set (80%) and a test set (20%). Produce summary statistics of the ###variable HiUse for the training and test sets separately.

library(caTools)  
  
set.seed(555)  
split = sample.split(atx$hi\_use, SplitRatio = 0.80)  
train = subset(atx, split == TRUE)  
test = subset(atx, split == FALSE)  
  
print(summary(train$hi\_use))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.3281 1.0000 1.0000

print(summary(test$hi\_use))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.3438 1.0000 1.0000

###(b) Fit a multiple logistic regression model predicting HiUse and using (among other things of your ###choice) Median.home.value as a predictor to the training set. Compute the test error – i.e. the ###fraction of the observations in the test set that are misclassified.

Based on this model, with a threshold of 50%, there were 0 false negatives and 4 false positives.Overall it was correct = of the time.

model <- glm(hi\_use~ customer\_class +median\_home\_value + median\_household\_income + median\_rent ,family=binomial(link='logit'),data=train)  
summary(model)

##   
## Call:  
## glm(formula = hi\_use ~ customer\_class + median\_home\_value + median\_household\_income +   
## median\_rent, family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.90422 -0.00005 -0.00003 0.57204 1.26161   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -2.438e+01 3.315e+03 -0.007 0.994  
## customer\_classIrrigation\_Residential -9.723e-01 4.487e+03 0.000 1.000  
## customer\_classMulti\_Family 2.139e+01 3.315e+03 0.006 0.995  
## customer\_classResidential 2.253e+01 3.315e+03 0.007 0.995  
## median\_home\_value -3.331e-06 3.969e-06 -0.839 0.401  
## median\_household\_income 8.787e-06 2.089e-05 0.421 0.674  
## median\_rent 4.006e-03 3.095e-03 1.295 0.195  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 143.297 on 108 degrees of freedom  
## Residual deviance: 58.294 on 102 degrees of freedom  
## (19 observations deleted due to missingness)  
## AIC: 72.294  
##   
## Number of Fisher Scoring iterations: 19

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

p <- predict(model, test, type = "response")  
print(summary(p))

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.0000 0.0000 0.6209 0.4759 0.8221 0.9818 9

p\_class <- ifelse(p > 0.5, "Predict 1", "Predict 0")  
print(table(p\_class, test[["hi\_use"]]))

##   
## p\_class 0 1  
## Predict 0 9 0  
## Predict 1 3 11

###(c) Redo the split so that you have different observations in the training and test sets and produce ###summary statistics of the variable HiUse in the new training and test sets. Fit the same multiple ###logistic regression model in (b) to the new training set and compute the new test error. ###Comment on your results with respect to the results from (a) and (b).

set.seed(2)  
split = sample.split(atx$hi\_use, SplitRatio = 0.80)  
train = subset(atx, split == TRUE)  
test = subset(atx, split == FALSE)  
print(summary(train$hi\_use))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.3281 1.0000 1.0000

print(summary(test$hi\_use))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.3438 1.0000 1.0000

model <- glm(hi\_use~ customer\_class +median\_home\_value + median\_household\_income + median\_rent ,family=binomial(link='logit'),data=train)  
summary(model)

##   
## Call:  
## glm(formula = hi\_use ~ customer\_class + median\_home\_value + median\_household\_income +   
## median\_rent, family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.06220 -0.00005 -0.00004 0.53758 1.68734   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -2.169e+01 3.407e+03 -0.006 0.995  
## customer\_classIrrigation\_Residential -2.231e-01 4.831e+03 0.000 1.000  
## customer\_classMulti\_Family 2.121e+01 3.407e+03 0.006 0.995  
## customer\_classResidential 2.261e+01 3.407e+03 0.007 0.995  
## median\_home\_value -3.827e-06 3.685e-06 -1.039 0.299  
## median\_household\_income 3.376e-05 2.380e-05 1.418 0.156  
## median\_rent 4.472e-05 2.240e-03 0.020 0.984  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 139.551 on 104 degrees of freedom  
## Residual deviance: 54.436 on 98 degrees of freedom  
## (23 observations deleted due to missingness)  
## AIC: 68.436  
##   
## Number of Fisher Scoring iterations: 19

p <- predict(model, test, type = "response")  
print(summary(p))

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.0000 0.0000 0.5733 0.4218 0.8100 0.9603 5

p\_class <- ifelse(p > 0.5, "Predict 1", "Predict 0")  
print(table(p\_class, test[["hi\_use"]]))

##   
## p\_class 0 1  
## Predict 0 12 0  
## Predict 1 4 11