Assignment 2: Classification

Ethan Tenison

2/26/2020

# Q1: Classification

Residential solar photovoltaics (PV) – the roof-mounted panels that generate electricity – are spreading throughout Austin. Two key drivers of the individual adoption decision are Wealth in K$, and Attitude on a scale beween 1-4. Suppose you collect data on a group of Austin homeowners where = Wealth in kdollars = Attitude, and Y = whether homeowners adopt solar PV. We fit a logistic regression and produce estimated coefficients β0 = −6, β1 = 0.05, and β2 = 1.

#### (a) Write out the function.

= + +

#### (b) Estimate the probability that a homeowner with Wealth of $40k and an Attitude of 3.5 adopts

solar PV.

= + + = + + = = = = = %

#### (c) What Wealth value would the homeowner in part (b) need to have in order to have a 50%

chance of adopting solar PV?

= + + = + + = + + = =

#### (d) What if you added a qualitative variable Tesla\_Owner as X3, that indicated whether the

household owns a Tesla plug-in electric car. Tesla cars are quite expensive, and those with solar PV systems benefit considerably from not paying for fuel. Write out the new function. What value might you expect for new Tesla\_Owner coefficient (β3) in this new function? How might you expect β1 and β2 in the NEW function to be different from β1 and β2 in the OLD function?

will probably be much larger than and because Tesla owners benefit from photovoltaics. The value for will probably decrease slightly because wealth is correlated with owning a Tesla, and some variation previously assigned to will be included in . It is unlikely that will change because attitude towards photovoltacics might not be associated with Tesla ownership.

#### (e) What is missing from this (admittedly simplified) analysis?

We might include an interaction variable for and . You might also want to include a variable for tax rebates. Not all homeowners may qualify, and that could have a big impact on their decision. Another variable that might be important is the avg price of PV panels, which have been declining.

# Q2: Classification in data (55 points)

Water use per person in Austin is currently at an historic low, but it can still be an issue – particularly when we are experience a dry, Texas summer. Please use the Austin Water - Residential Water Consumption data (<https://data.austintexas.gov/Utilities-and-City-Services/Austin-Water-Residential-WaterConsumption/sxk7-7k6z>) for this question. Take note – this is a real world data set and as such may not be clean.

library(dplyr)  
library(janitor)  
library(reshape2)  
df <- read.csv("Austin\_Water\_-\_Residential\_Water\_Consumption.csv")  
df <- clean\_names(df)  
df$year <- substr(df$year\_month, 1, 4)  
df$year <- as.numeric(df$year)  
df$month <- substr(df$year\_month, 5, 6)  
df$month <- as.factor(df$month)  
df <- dplyr::select(df, -c(year\_month))  
df$customer\_class <- gsub(" - ", "\_", df$customer\_class)  
df <- filter(df, postal\_code != "")  
df$total\_gallons[is.na(df$total\_gallons)] <- 0  
dfwide <- dcast(data=df, postal\_code+month+year~customer\_class, value.var= 'total\_gallons')  
dfwide <- clean\_names(dfwide)  
df$total\_gallons[is.na(df$total\_gallons)] <- 0  
dfwide[is.na(dfwide)] <- 0

#### (a) Produce some numerical and graphical summaries of the Water data. What patterns do you

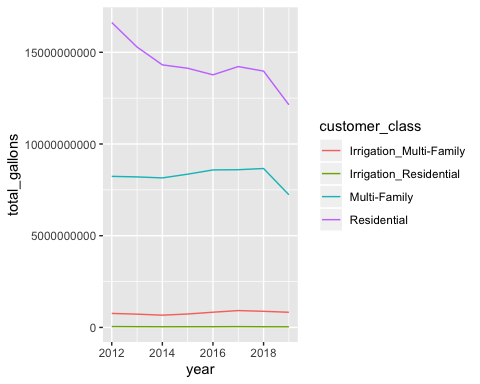
see?

Total water consumption declined after water restrictions were introduced in 2018. Irrigation remained basically the same even though these numbers should have declined as well.

library(ggplot2)  
options(scipen=999)  
print(summary(dfwide))

## postal\_code month year irrigation\_multi\_family  
## 78613 : 94 06 : 377 Min. :2012 Min. : 0   
## 78617 : 94 07 : 377 1st Qu.:2013 1st Qu.: 28050   
## 78652 : 94 08 : 377 Median :2015 Median : 565300   
## 78653 : 94 09 : 377 Mean :2015 Mean : 1434146   
## 78660 : 94 10 : 377 3rd Qu.:2017 3rd Qu.: 1923350   
## 78701 : 94 01 : 376 Max. :2019 Max. :24298000   
## (Other):3858 (Other):2161   
## irrigation\_residential multi\_family residential   
## Min. : 0 Min. : 0 Min. : 0   
## 1st Qu.: 0 1st Qu.: 2370450 1st Qu.: 7162150   
## Median : 12450 Median :11954550 Median : 19374950   
## Mean : 71293 Mean :14931954 Mean : 25883596   
## 3rd Qu.: 45675 3rd Qu.:19102750 3rd Qu.: 39344625   
## Max. :2704000 Max. :92554300 Max. :125645900   
##

total\_water <- df %>% group\_by(year, customer\_class) %>%   
 summarize(total\_gallons = sum(total\_gallons))  
  
tw <- ggplot(total\_water, aes(x=year, y=total\_gallons, group=customer\_class, color=customer\_class)) +  
 geom\_line()  
print(tw)



#### (b) Create a binary variable, HiResIrr, that contains a 1 if Irrigation\_Residential contains a value

above its mean, and a 0 if Irrigation\_Residential contains a value below its mean. Use the full data set to perform a logistic regression with HiResIrr as the response and other variables as predictors (besides the original Irrigation\_Residential variable). Provide a summary of your obtained results. Do any of the predictors appear to be statistically significant? If so, which ones? Does it look like residential irrigation has decreased over the past few years?

None of the postal codes were statistically significant, so I removed them.Year, multi-family, and residential were significant and had a positive impact on the probability of HiResIrr being above the mean. For every one unit increase in year is associated with an increase in the log-odds of being above the mean by 0.055649003832. When I converted month to a factor, only the hottest months of the year are significant in the positive direction. Irrigation\_multifamily was significant in the negative direction.

library(caTools)  
logdf <- dfwide  
  
for (i in 1:length(logdf$irrigation\_residential)){  
 if(logdf$irrigation\_residential[i] > mean(logdf$irrigation\_residential)){   
 logdf$HiResIrr[i] <- 1   
 }  
 else if(logdf$irrigation\_residential[i] <= mean(logdf$irrigation\_residential)){   
 logdf$HiResIrr[i] <- 0   
 }  
}  
set.seed(88)  
split = sample.split(logdf$HiResIrr, SplitRatio = 0.75)  
logdf <- dplyr::select(logdf, -c(irrigation\_residential, postal\_code))  
train = subset(logdf, split == TRUE)  
test = subset(logdf, split == FALSE)  
  
model <- glm(HiResIrr ~.,family=binomial(link='logit'),data=train)  
summary(model)

##   
## Call:  
## glm(formula = HiResIrr ~ ., family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1999 -0.5570 -0.3719 -0.2499 2.8841   
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) -115.962396952931 47.535119512738 -2.440  
## month02 0.283263898648 0.287384733046 0.986  
## month03 0.268087791079 0.292944928637 0.915  
## month04 0.354008960996 0.284999727342 1.242  
## month05 0.822327630847 0.274118138173 3.000  
## month06 0.738722950314 0.277826836533 2.659  
## month07 1.184888739088 0.270106101184 4.387  
## month08 1.531516976149 0.268954742029 5.694  
## month09 1.396497954700 0.271889900322 5.136  
## month10 1.235925304046 0.269595393645 4.584  
## month11 0.801228533561 0.287030089686 2.791  
## month12 0.521886160942 0.291221768812 1.792  
## year 0.055649003832 0.023577081824 2.360  
## irrigation\_multi\_family -0.000000426759 0.000000036306 -11.755  
## multi\_family 0.000000027343 0.000000003148 8.687  
## residential 0.000000049457 0.000000002667 18.543  
## Pr(>|z|)   
## (Intercept) 0.01471 \*   
## month02 0.32430   
## month03 0.36011   
## month04 0.21419   
## month05 0.00270 \*\*   
## month06 0.00784 \*\*   
## month07 0.0000115055 \*\*\*  
## month08 0.0000000124 \*\*\*  
## month09 0.0000002803 \*\*\*  
## month10 0.0000045536 \*\*\*  
## month11 0.00525 \*\*   
## month12 0.07312 .   
## year 0.01826 \*   
## irrigation\_multi\_family < 0.0000000000000002 \*\*\*  
## multi\_family < 0.0000000000000002 \*\*\*  
## residential < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3055.8 on 3315 degrees of freedom  
## Residual deviance: 2385.8 on 3300 degrees of freedom  
## AIC: 2417.8  
##   
## Number of Fisher Scoring iterations: 5

#### (c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the

confusion matrix is telling you about the types of mistakes made by logistic regression.

Based on this model, with a threshold of 50%, there is a high number of false positives relative to true positives.Overall it was correct = of the time.

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

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.01321 0.05412 0.10028 0.17570 0.22754 0.97395

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

##   
## p\_class 0 1  
## Predict 0 887 144  
## Predict 1 27 48

#### (d) Split the data into a training set (80%) and a test set (20%). Perform KNN with several values of K

and all the variables from (b) in order to predict HiResIrr. What test errors do you obtain? Which value of K seems to perform the best on this data set? Comment on the sensitivity and specificity

The highest overall prediction rate was reached at k=5. The sensitivity is, or true positive rate is , while the specificity, or true negative is relatively higher at

library(class)  
library(gmodels)  
  
dfknn <- dfwide  
for (i in 1:length(dfknn$irrigation\_residential)){  
 if(dfknn$irrigation\_residential[i] > mean(dfknn$irrigation\_residential)){   
 dfknn$HiResIrr[i] <- 1   
 }  
 else if(dfknn$irrigation\_residential[i] <= mean(dfknn$irrigation\_residential)){   
 dfknn$HiResIrr[i] <- 0   
 }  
}  
  
set.seed(88)  
dfknn <- dplyr::select(dfknn, -c(irrigation\_residential, postal\_code))  
dmy <- dummyVars(" ~ .", data = dfknn)  
dfknn <- data.frame(predict(dmy, newdata = dfknn))  
split = sample.split(dfknn$HiResIrr, SplitRatio = 0.8)  
train\_knn = subset(dfknn, split == TRUE)  
train\_labels = train\_knn[["HiResIrr"]]  
train\_knn[is.na(train\_knn)] <- 0  
  
test\_knn = subset(dfknn, split == FALSE)  
test\_labels = test\_knn[["HiResIrr"]]  
test\_knn[is.na(test\_knn)] <- 0  
  
  
dfknn\_prediction = class::knn(train= train\_knn, test = test\_knn, cl= train\_labels, k = 4)  
CrossTable(x= test\_labels, y= dfknn\_prediction, chisq = FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 884   
##   
##   
## | dfknn\_prediction   
## test\_labels | 0 | 1 | Row Total |   
## -------------|-----------|-----------|-----------|  
## 0 | 684 | 47 | 731 |   
## | 7.883 | 41.542 | |   
## | 0.936 | 0.064 | 0.827 |   
## | 0.921 | 0.333 | |   
## | 0.774 | 0.053 | |   
## -------------|-----------|-----------|-----------|  
## 1 | 59 | 94 | 153 |   
## | 37.665 | 198.478 | |   
## | 0.386 | 0.614 | 0.173 |   
## | 0.079 | 0.667 | |   
## | 0.067 | 0.106 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 743 | 141 | 884 |   
## | 0.840 | 0.160 | |   
## -------------|-----------|-----------|-----------|  
##   
##

####(e) Perform LDA on the training data in order to predict HiResIrr using all the variables from (b) as predictors. What is the test error of the model obtained? Comment on the sensitivity and specificity.

The overall test error is .The Sensitivity of the LDA function is high at , but the specificity is low at .

library(MASS)  
  
fit <- lda(HiResIrr ~ year + irrigation\_multi\_family + multi\_family + residential, data=train)  
  
pred.train <- predict(fit,train)$class  
pred.test <- predict(fit,test)$class  
  
mean(pred.train == train$HiResIrr)

## [1] 0.8477081

mean(pred.test == test$HiResIrr)

## [1] 0.835443

confusionMatrix(pred.test, reference = as.factor(test$HiResIrr))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 871 139  
## 1 43 53  
##   
## Accuracy : 0.8354   
## 95% CI : (0.8123, 0.8568)   
## No Information Rate : 0.8264   
## P-Value [Acc > NIR] : 0.2265   
##   
## Kappa : 0.2853   
##   
## Mcnemar's Test P-Value : 0.000000000001897  
##   
## Sensitivity : 0.9530   
## Specificity : 0.2760   
## Pos Pred Value : 0.8624   
## Neg Pred Value : 0.5521   
## Prevalence : 0.8264   
## Detection Rate : 0.7875   
## Detection Prevalence : 0.9132   
## Balanced Accuracy : 0.6145   
##   
## 'Positive' Class : 0   
##

#### (f) Perform QDA on the training data in order to predict HiResIrr using all the variables from (b) as

predictors. What is the test error of the model obtained?

The test error for the QDA function was slightly higher than LDA at

fit <- qda(HiResIrr ~ year + irrigation\_multi\_family + multi\_family + residential, data=train)  
  
pred.train <- predict(fit,train)$class  
pred.test <- predict(fit,test)$class  
  
mean(pred.train == train$HiResIrr)

## [1] 0.8443908

mean(pred.test == test$HiResIrr)

## [1] 0.8291139

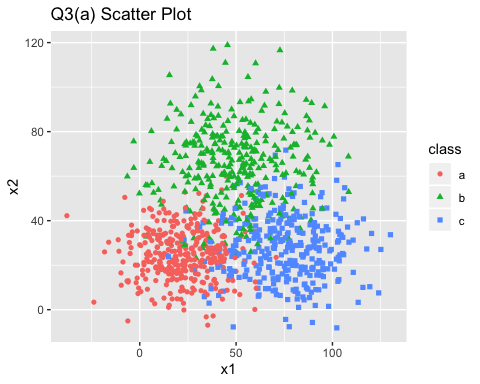
confusionMatrix(pred.test, reference = as.factor(test$HiResIrr))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 864 139  
## 1 50 53  
##   
## Accuracy : 0.8291   
## 95% CI : (0.8056, 0.8509)  
## No Information Rate : 0.8264   
## P-Value [Acc > NIR] : 0.4246   
##   
## Kappa : 0.2709   
##   
## Mcnemar's Test P-Value : 0.0000000001543   
##   
## Sensitivity : 0.9453   
## Specificity : 0.2760   
## Pos Pred Value : 0.8614   
## Neg Pred Value : 0.5146   
## Prevalence : 0.8264   
## Detection Rate : 0.7812   
## Detection Prevalence : 0.9069   
## Balanced Accuracy : 0.6107   
##   
## 'Positive' Class : 0   
##

# Q3: Classification in simulated data (20 points)

####(a) Create a data frame with Index from 1-1000. In a variable called Class, randomly assign each row one of c(“a”,”b”,”c”). Draw values of X1 and X2 for each class according to the following: Class a: X1~N(20, 17), X2~N(25, 12), Class b: X1~N(50, 22), X2~N(65, 19), Class c: X1~N(75, 20), X2~N(27, 15). Plot the data as points with X1 on the X-axis, X2 on the Y-axis, and the color determined by Class (hint: use ggplot). Draw – by hand – your estimate of the Bayes Decision Boundaries.

library(ggplot2)  
let <- letters[1:3]  
df <- data.frame("index" = 1:1000, "class" = sample(let, replace = TRUE, size = 1000))  
  
for (i in 1:length(df$index)){  
 if(df$class[i] == "a"){   
 df$x1[i] <- rnorm(1, 20, 17)  
 df$x2[i] <- rnorm(1, 25, 12)  
 }  
 else if(df$class[i] == "b"){   
 df$x1[i] <- rnorm(1, 50, 22)  
 df$x2[i] <- rnorm(1, 65, 19)  
 }  
 else if(df$class[i] == "c"){   
 df$x1[i] <- rnorm(1, 75, 20)  
 df$x2[i] <- rnorm(1, 27, 15)  
 }  
}  
  
  
  
ggplot(df, aes(x=x1, y=x2, shape = class, color = class)) + geom\_point() + ggtitle("Q3(a) Scatter Plot")



#### (b) Use LDA to predict class based on X1 and X2. Plot the predicted classes just as you did in (a).

Comment on the differences between the two plots in Q3, and evaluate your hand drawn Bayes Decision Boundaries.

require(MASS)  
require(ggplot2)  
require(scales)

## Loading required package: scales

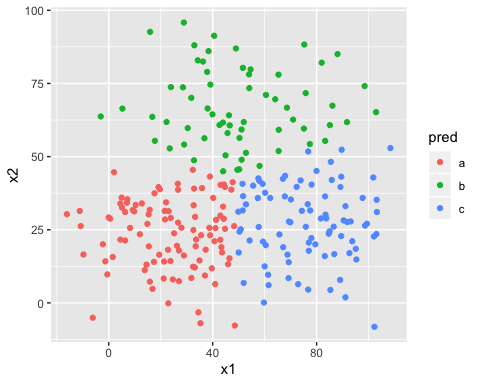
require(gridExtra)

## Loading required package: gridExtra

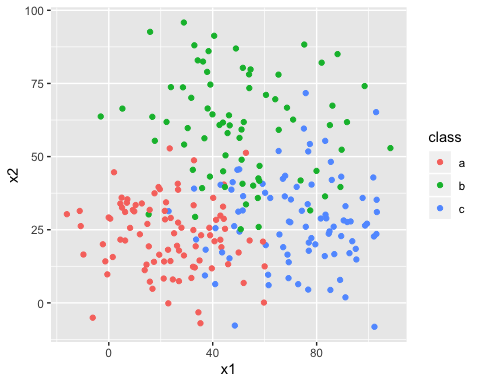
##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

set.seed(88)  
split = sample.split(df, SplitRatio = 0.75)  
train = subset(df, split == TRUE)  
test = subset(df, split == FALSE)  
  
  
fit <- lda(class ~ x1 + x2, data=train)  
pred <- predict(fit, test)$class  
ggplot(data = test, aes(x1, x2, color=pred)) + geom\_point()



ggplot(data = test, aes(x1,x2, color=class)) + geom\_point()



ldacorrect = mean(pred == df$class)  
ldaerror = 1- ldacorrect  
ldaerror

## [1] 0.652