Allan Swartz
6/9/16

Classification of PG&E's Customers by Electricity Usage

Step 1 – Collecting Data. The premise of this investigation involves taking data involving electricity usage and trying to classify the customer type based on these values. The data was collected by PG&E's own data team and is available to the public on their site. The period of data collection was done in the fourth quarter of 2015.

Step 2 – Exploring and Preparing the Data. The dataset contains numerous variables, some of which are important and some of which are not necessary to assist with classification. First, it is best to get a feel for what to expect in the dataset.

```
'data.frame':   7733 obs. of  8 variables:
 $ ZipCode       : int  93101 93101 93105 93105 93110 93110 93117 93117 93201 93202 ...
 $ Month         : int  10 10 10 10 10 10 10 10 10 10 ...
 $ Year          : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ CustomerClass : chr  "Elec- Commercial" "Elec- Residential" "Elec- Commercial" "Elec-
Residential" ...
 $ Combined      : chr  "Y" "Y" "Y" "Y" ...
 $ TotalCustomers: num  0 0 0 0 0 0 0 0 311 0 ...
 $ TotalkWh      : num  0 0 0 0 0 ...
 $ AveragekWh    : num  NA NA NA NA NA NA NA NA 548 NA ...
```

There are a total of 8 variables. Each observation is an individual California Zip Code, including the month, year, the class of customer, how many of those customers reside in the area, and electricity usage measurements. The Combined column indicates PG&E's method of grouping, in which very small groups of customers are allocated into neighboring zip codes, hence the missing values and zero usage.

The goal now is to use variables to attempt to predict what kind of class a customer is solely based on electricity usage and household count without any information of their location in California. While zip codes are no longer required to classify in this way, other variables need to be examined.

There were a few actions needed to be taken to clean the data before modelling could take place. Eliminating any observations that were combined with other ones does not actually remove customer counts, so they were filtered out as they provided no additional information. The entire set was of the year 2015, so this column was removed as well. Month should have no bearing in the process since each observation appears in every month. The customer class variable was slightly revised to leave out the electricity part of the name; all observations were based on electricity.

Here is the data once again after all cleaning is finished.

```
CustomerClass      TotalCustomers     TotalkWh          AveragekWh
 Length:2889        Min.   :   18   Min.   :    7816   Min.   :   98
 Class :character   1st Qu.:  376   1st Qu.:  400641   1st Qu.:  524
 Mode  :character   Median : 1292   Median : 2109704   Median :  660
                    Mean   : 4519   Mean   : 3496635   Mean   : 1992
                    3rd Qu.: 7320   3rd Qu.: 5633483   3rd Qu.:  964
```

```
                  Max.   :27819   Max.   :44948850   Max.   :57647

Agricultural   Commercial  Residential
 0.004153686  0.188300450  0.807545864
```

Customers and electricity usage seem to have extremely high variability. This actually may be useful in helping the algorithm of choice to divide the data and classify it easier. One potential problem is that Agricultural customers make up less than 1% of the dataset. With 2889 observations, that is an extremely small subset. The biggest challenge may be trying to correctly group those few customers into their respective category.

Step 3 – Training the Model. The first model that will be used on the data is kNN. However, due to the wide range of values and the nature of the algorithm, it is necessary to first scale everything down before training the model.

```
TotalCustomers       TotalkWh          AveragekWh           CustomerClass
 Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Agricultural:  12
 1st Qu.:0.01288   1st Qu.:0.008741   1st Qu.:0.007402   Commercial  : 544
 Median :0.04583   Median :0.046770   Median :0.009766   Residential :2333
 Mean   :0.16189   Mean   :0.077631   Mean   :0.032910
 3rd Qu.:0.26265   3rd Qu.:0.125179   3rd Qu.:0.015048
 Max.   :1.00000   Max.   :1.000000   Max.   :1.000000
```

The normalization was done using the minimum and maximum scaling technique. Distances computed will no longer have variables dominating the others.

The dataset will then be split into two sets, one for training and validation and the other to be used for testing. It is critical that the proportion of customer classes in either one are as close as possible to produce reasonable results.

```
Agricultural   Commercial  Residential
 0.004610951  0.188472622  0.806916427

Agricultural   Commercial  Residential
 0.004610951  0.188472622  0.806916427
```

The training and validation set is comprised of 60% of the observations, while the test set will be the remaining 40%. The two tables show the proportions for each of these sets, and the consistency between them is preserved.

```
k-Nearest Neighbors

1735 samples
   3 predictor
   3 classes: 'Agricultural', 'Commercial', 'Residential'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1562, 1561, 1562, 1562, 1562, 1562, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   3  0.9913627  0.9722114
   5  0.9913594  0.9722930
   7  0.9925121  0.9758386
```

```
   9  0.9919341  0.9740620
  11  0.9913561  0.9724065
  13  0.9902033  0.9686237
  15  0.9890506  0.9649298

Kappa was used to select the optimal model using  the one SE rule.
The final value used for the model was k = 11.
```

Using a 10-fold cross-validation method of sampling as well as various values of k, it was determined that k=11 provides the most efficient model. Indeed, it has an impressive accuracy of 99% and a Kappa of 0.9686, the metric used to determine model performance.

Step 4 – Evaluating Model Performance. Even with the high accuracy of the model, it is not ready to be considered the best possible model until it is put to use on another dataset. As such, the test set will be used to determine this.

```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  1154


                     | Predicted Customer Class
Actual Customer Class |  Commercial | Residential |   Row Total |
----------------------|-------------|-------------|-------------|
         Agricultural |           3 |           1 |           4 |
                      |       0.003 |       0.001 |             |
----------------------|-------------|-------------|-------------|
           Commercial |         214 |           3 |         217 |
                      |       0.185 |       0.003 |             |
----------------------|-------------|-------------|-------------|
          Residential |           0 |         933 |         933 |
                      |       0.000 |       0.808 |             |
----------------------|-------------|-------------|-------------|
         Column Total |         217 |         937 |        1154 |
----------------------|-------------|-------------|-------------|
```

Comparing the predicted values of the test set with the actual values contained in the customer class column, the results do reasonably well here. The accuracy is consistent with the statement made during the modelling process that it is 99%. The issue as presented before is evident in the table however, in that Agricultural is thrown into other categories. In fact, there are only 4 total observations of this class, so it is extremely difficult to differentiate these between the others. There isn't enough observations in the first place to find any patterns between them.

This only uses a distance measurement to classify, so it is worth trying out another method to see if any improvements can be made.

Step 5 – Improving the Model. To look for any improvements to classifying this set, a new kind of process will be used. Specifically, a decision-based tree model using the C5.0 algorithm will be performed to see if different results can be achieved.

```
C5.0

1735 samples
   3 predictor
   3 classes: 'Agricultural', 'Commercial', 'Residential'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 1562, 1561, 1561, 1561, 1562, 1562, ...
Resampling results across tuning parameters:

  model  winnow  trials  Accuracy   Kappa
  rules  FALSE    1      0.9906129  0.9699911
  rules  FALSE   10      0.9917068  0.9734516
  rules  FALSE   20      0.9914188  0.9725003
  rules   TRUE    1      0.9898631  0.9675032
  rules   TRUE   10      0.9912441  0.9718341
  rules   TRUE   20      0.9909584  0.9709915
  tree   FALSE    1      0.9903824  0.9692290
  tree   FALSE   10      0.9921673  0.9749126
  tree   FALSE   20      0.9918231  0.9738277
  tree    TRUE    1      0.9896326  0.9667405
  tree    TRUE   10      0.9911301  0.9714931
  tree    TRUE   20      0.9906700  0.9700731

Kappa was used to select the optimal model using  the largest value.
The final values used for the model were trials = 10, model = tree and winnow
 = FALSE.
```

Instead of one 10-fold cross-validation, it was extended to be repeated 10 times using this method in order to improve model performance.

Going back to the unscaled data, as decision trees do not need this to be successful, the best fitted model was now automatically found to be a tree model with 10 trials and a winnow of False. The accuracy is once again 99% and the Kappa value is 0.9749. Already this is looking slightly better than the model found through kNN, but as before it is best to see results using the test set.

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  1154


                     | Predicted Customer Class
Actual Customer Class | Agricultural |   Commercial |  Residential |    Row Total |
```

Agricultural	1	2	1	4
	0.250	0.500	0.250	0.003
	0.001	0.002	0.001	
Commercial	0	214	3	217
	0.000	0.986	0.014	0.188
	0.000	0.185	0.003	
Residential	0	1	932	933
	0.000	0.001	0.999	0.808
	0.000	0.001	0.808	
Column Total	1	217	936	1154

The accuracy found using the C5.0 algorithm is identical to the previous one. However this could be considered an improvement only because it correctly identified one of the Agricultural customers. Even though this is still not the majority of them, there is at least evidence that it can correctly classify this difficult category.

It does appear that Commercial and Agriculture get mixed up, whereas Residential appears different enough from either of them so as not to get mixed together. One reason classifying this particular dataset is useful is to determine potential electricity usage rates depending on how they are labeled. Thresholds could be set based on the number of customers and usage, and rates adjusted accordingly for any given region's usage. Without even inspecting location, these details could be chosen using the decision tree model. There may be some contention between Agricultural and Commercial customers, as they may potentially be mixed together, so careful consideration of the cost of incorrectly labeling an Agricultural customer as a Commercial one should be exercised. Overall, the majority of customers are either Commercial or Residential, and this model performs well enough to distinguish between the two of them.

Appendix:

```
## Project 2 ##

##fourth quarter totals of electricy usage
pge <- read.csv("PGE_2015_Q4_ElectricUsageByZip.csv",stringsAsFactors=F)

#general picture of the data
str(pge)

#using combined, filter out missing data and remove unncessary variables
pge.rev <- pge[!is.element(pge$Combined,"Y"),]
pge.rev <- pge.rev[,-c(1:3,5)]

#remove part of the string
pge.rev$CustomerClass <- sapply(strsplit(pge.rev$CustomerClass,split="- ",
                                fixed=T),function(x)(x[2]))

#the breakdown of variables to be classified
summary(pge.rev)
prop.table(table(pge.rev$CustomerClass))

##subsetting a training and validation set
#normalize to prepare for knn
normalize <- function(x){return((x-min(x))/(max(x)-min(x)))}
pge.n <- as.data.frame(lapply(pge.rev[2:4],normalize))
pge.n <- cbind(pge.n,pge.rev$CustomerClass)
colnames(pge.n) <- c("TotalCustomers","TotalkWh","AveragekWh","CustomerClass")
summary(pge.n)

#split dataset into 60 percent train 40 percent test
train.sub <- createDataPartition(pge.n$CustomerClass,p=0.60,list=F)
train.pge <- pge.n[train.sub,]
test.pge <- pge.n[-train.sub,]

prop.table(table(train.pge$CustomerClass))
prop.table(table(train.pge$CustomerClass))

##modelling and evaluating
#model with knn
ctrl <- trainControl(method="cv",number=10,selectionFunction="oneSE")
grid <- expand.grid(.k=c(3,5,7,9,11,13,15))

model <- train(CustomerClass~.,data=train.pge,method="knn",metric="Kappa",
               trControl=ctrl,tuneGrid=grid)
model

p <- predict(model,test.pge)
CrossTable(x=test.pge$CustomerClass,y=p,prop.chisq=F,prop.c=F,prop.r=F,
           dnn=c("Actual Customer Class","Predicted Customer Class"))

##improving the model
#using C5.0 decision trees
improved.train.pge <- pge.rev[train.sub,]
improved.test.pge <- pge.rev[-train.sub,]

ctrl <- trainControl(method="repeatedcv",number=10,repeats=10)
improved.model <- train(CustomerClass~.,data=improved.train.pge,method="C5.0",
```

```
                          metric="Kappa",trControl=ctrl)
improved.model

improved.p <- predict(improved.model,improved.test.pge)
CrossTable(x=improved.test.pge$CustomerClass,y=improved.p,prop.chisq=F,
           prop.c=F,dnn=c("Actual Customer Class","Predicted Customer Class"))
```