# Project 2

**Decision & Regression Trees**

Brooke O'Donnell

Fall Semester 2020: Data & Text Mining

October 11th, 2020

Part A:

Build decision trees to classify the customers into the two categories in the "Churn" variable using the "Churn.csv" data set. The "Churn" variable is a binary categorical variable (factor) showing whether a customer has "churned" or not. "Churn" means terminating the subscription or leaving the company. Please include the following variables in your subset: gender, SeniorCitizen, Partner, Dependents, tenure, MonthlyCharges, Contract, and Churn. You need to wrangle your data into a format proper for decision tree building.

Part B:

For this portion of the project you need the data set "mortgagedefaul.xlsx." You need to figure out how to read the file into R. What is particularly challenging is that the data you need is on the second sheet of the EXCEL file, while the first sheet is the code book, which contains information of the variables in the data set.To accomplish the required tasks, you need the variables below. Bo_Age, Credit_score, First_home, Tot_mthly_incm, pur_prc_amt, orig_apprd_val_amt, Median_state_inc, Orig_LTV_Ratio_Pct, Ln_Orig

**Part A:**

1. What is the Gini impurity measure if we were to predict the values of "Churn" according to the greater proportion? (i.e. without using any predictor or feature)

```
churn.prop<-prop.table(table(subset1$Churn))

head(churn.prop)

Yes.churn  No.churn
0.2653699 0.7346301

G<-1-(0.2653699^2+0.7346301^2)
```

- Without classifying the cases, we have two categories: 27% yes.churn and 73% no.churn. Without any feature variables the GINI impurity measure for 'churn' is .39

2. Please show that the variable "senior" leads to information gain if used as a feature (variable). (Hint: you need to calculate the two partial Gini indexes for "Senior" and "NonSenior" and then the weighted Gini index for the variable)

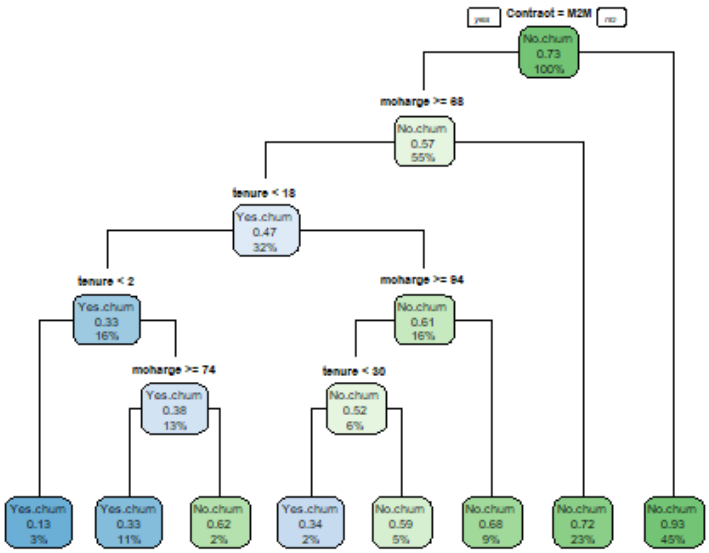
```
ginisen<-1-((476/(476+666))^2+(666/(476+666))^2)
ginisen
0.4861597
gininon.sen<-1-((1393/(1393+4508))^2+(4508/(1393+4508))^2)
gininon.sen
0.3606731
wginisen<-((1393+4508)/7043)*(0.3606731) + ((476+666)/7043)*0.4861597
round(wginisen, 3)
```

- If 'senior' was used as a feature variable customers classified as "Non-Seniors" would provide more information according to the lower GINI impurity measure of 0.3606731. This is compared to customers classified as "seniors" which does not provide much information because the GINI is 0.4861597. Customers classified as "seniors" are almost split evenly 1/2 and 1/2 because the GINI measure is closer to .5. The weighted GINI index for the variable 'senior' is .381.

3. Split the data set into training (70%) and testing set (30%); then build a decision tree for the target "Churn" by including all feature variables and setting the cp value to 0.01.

```
set.seed(123)
ind<-sample(2, nrow(subset1), replace=TRUE, prob=c(0.7, 0.3))
churn.train<-subset1[ind==1,]
churn.test<-subset1[ind==2,]
library(rpart)
myform<-Churn~gender+senior+Partner+Dependents+tenure+mcharge+Contract
churn.rpart<- rpart(myform,
            data=churn.train,
            method="class",
            minsplit=2,
            minbucket=1,
```

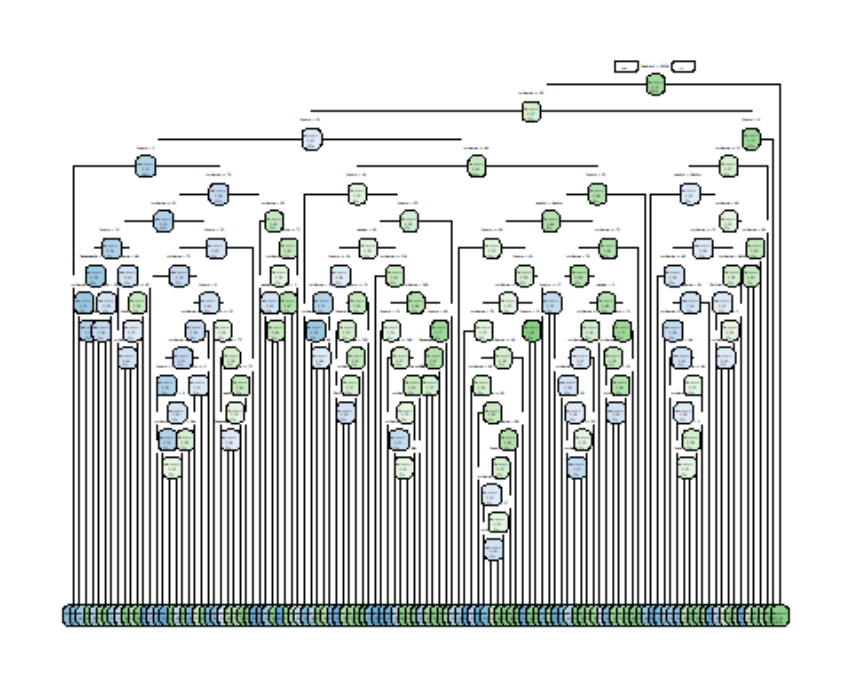
```
            cp=0.01)
print(churn.rpart)
rpart.plot(churn.rpart, type = 1, extra = "auto")
```



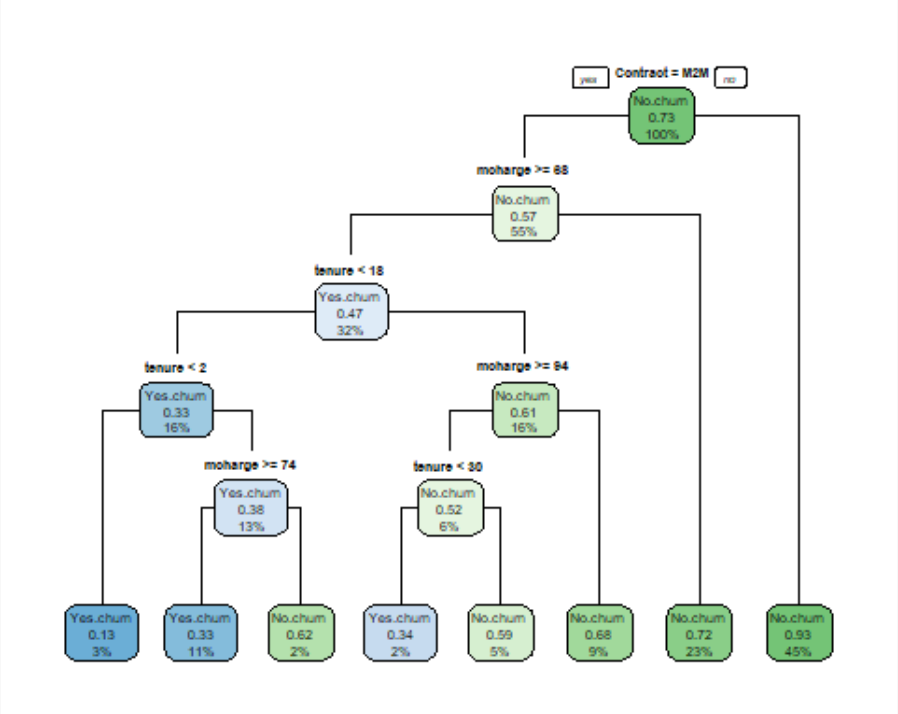

- How many feature variables are used in this decision tree?
  - There are three feature variables used in this decision tree
- What features are they?
  - The feature variables are contract, mcharge, and tenure. The targeted feature variable is Churn
- How many leaf nodes are there?
  - There are 8 leaf nodes in this decision tree

4. Try cp = 0.001, 0.0055. Discuss why one might be better than the other. Then obtain the Confusion Matrix for the two tree models with cp=0.001 and 0.0055. Based on the differences of the indexes of “accuracy”, “sensitivity,” and “specificity” in the two models, what are the advantages and disadvantages of the two decision trees?

```
rpart.plot(churn.rpart.001, type = 1, extra = "auto")
```



```
rpart.plot(churn.rpart.0055, type = 1, extra = "auto")
```




```
test.pred.001<-predict(churn.rpart.001, churn.test, type="class" )
CM.001<-confusionMatrix(test.pred.001, churn.test$Churn)
CM.001
Confusion Matrix and Statistics
```

```
          Reference
Prediction  Yes.churn No.churn
Yes.churn       256      170
No.churn        290     1377
Accuracy : 0.7802
Kappa : 0.3865
Mcnemar's Test P-Value : 2.883e-08
Sensitivity : 0.4689
Specificity : 0.8901
Balanced Accuracy : 0.6795
'Positive' Class : Yes.churn
printcp(churn.rpart.001)
Classification tree:
rpart(formula = myform, data = churn.train, method = "class",
minsplit = 2, minbucket = 1, cp = 0.001)
Variables actually used in tree construction:
Contract   Dependents gender     mcharge    Partner    senior     tenure    Root node error:
1323/4950 = 0.26727
         CP nsplit rel error  xerror     xstd
1  0.0692870      0   1.00000 1.00000 0.023534
2  0.0109599      3   0.79214 0.80726 0.021875
3  0.0102041      5   0.77022 0.81104 0.021912
4  0.0040312      7   0.74981 0.79819 0.021785
5  0.0035273     12   0.72336 0.77929 0.021595
6  0.0030234     15   0.71277 0.77551 0.021556
7  0.0022676     18   0.70370 0.77022 0.021502
8  0.0015117     24   0.69010 0.78005 0.021603
9  0.0012094     67   0.61905 0.79289 0.021733
10 0.0011338     72   0.61300 0.81557 0.021956
11 0.0010078     99   0.58050 0.82464 0.022044
12 0.0010000    110   0.56916 0.83220 0.022116
test.pred.0055<-predict(churn.rpart.0055, churn.test, type="class" )
CM.0055<-confusionMatrix(test.pred.0055, churn.test$Churn)
CM.0055
Confusion Matrix and Statistics
          Reference
Prediction  Yes.churn No.churn
Yes.churn       211      129
No.churn        335     1418
 Accuracy : 0.7783
Kappa : 0.3452
Mcnemar's Test P-Value : < 2.2e-16
Sensitivity : 0.3864
Specificity : 0.9166
Balanced Accuracy : 0.6515
‘Positive' Class : Yes.churn
printcp(churn.rpart.0055)
```

```
Classification tree:
rpart(formula = myform, data = churn.train, method = "class",
   minsplit = 2, minbucket = 1, cp = 0.0055)
Variables actually used in tree construction:
Contract mcharge  tenure
Root node error: 1323/4950 = 0.26727
n= 4950
        CP nsplit rel error  xerror     xstd
1 0.069287      0   1.00000 1.00000 0.023534
2 0.010960      3   0.79214 0.81784 0.021978
3 0.010204      5   0.77022 0.82237 0.022022
4 0.005500      7   0.74981 0.80121 0.021816
```

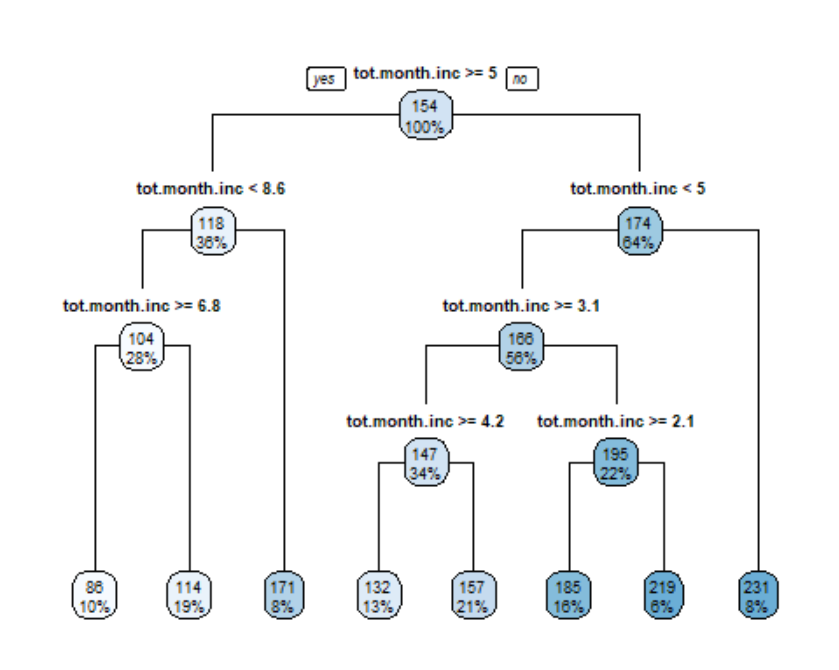
| **Confusion Matrix .0055** | | **Confusion Matrix .001** | |
|---|---|---|---|
| Accuracy | 0.7783 | Accuracy | 0.7802 |
| Specificity | 0.9166 | Specificity | 0.8901 |
| Sensitivity | 0.3864 | Sensitivity | 0.4689 |
| R Squared (1-Relative Error) | 0.25019 | R Squared (1-Relative Error) | 0.43084 |
| Advantages<br>• Model is simpler<br>• Larger Specificity: capacity to detect the negatives (the people who churned)<br>• **Accepted model compared to the** churn.rpart.001 **model** | | Advantages<br>• Higher r-square<br>• Slightly higher accuracy | |
| Disadvantages<br>• Lower r-square<br>• Lower Sensitivity | | Disadvantages<br>• Model is not simple, runs into the problem of overfitting<br>• Too complex, can't be interpreted<br>• **Rejected model compared to the** churn.rpart.0055 **model** | |

**Part B:**

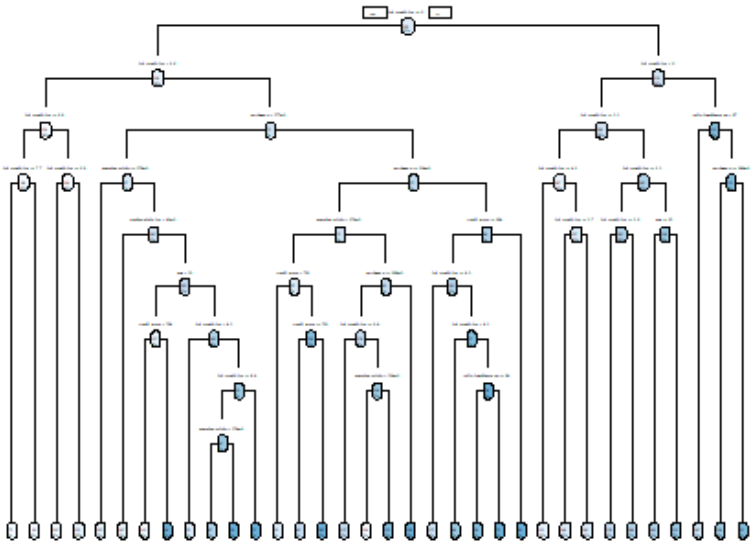
1. Please split the data set into training (70%) and testing sets (30%) and then build a regression tree using the training set. The regression tree is to be built to predict the value of loan amount (Ln_Orig) by including all the other variables as features and setting "cp" values at 0.01, 0.005, and 0.001 respectively. From the three summary outputs, what cp value is optimal for the tree model? (You only need to show information relevant to your assessment. Please do not include long output!)

```
set.seed(123)
ind<-sample(2, nrow(subset2), replace=TRUE, prob=c(0.7, 0.3))
loan.train<-subset2[ind==1,]
loan.test<-subset2[ind==2,]
```

```
mytree.part.2<- rpart(loan.value~.,
            data=loan.train,
            method="anova",
            minsplit=2,
            minbucket=1,
            cp=0.01)
mytree.part.2
                        rpart.plot(mytree.part.2, type = 1, extra = "auto")
```

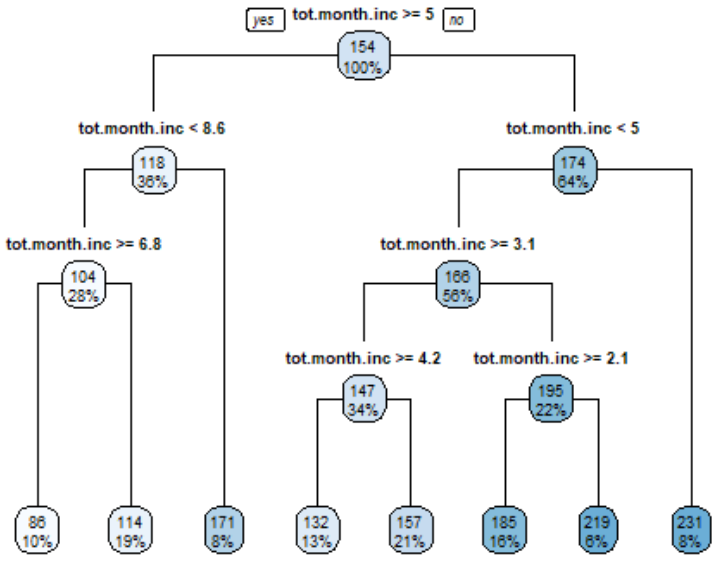



```
mytree.part.2.001<- rpart(loan.value~.,
            data=loan.train,
            method="anova",
            minsplit=2,
            minbucket=1,
            cp=0.001)
mytree.part.2.001
tree.001<-rpart.plot(mytree.part.2.001, type = 1, extra = "auto")
```



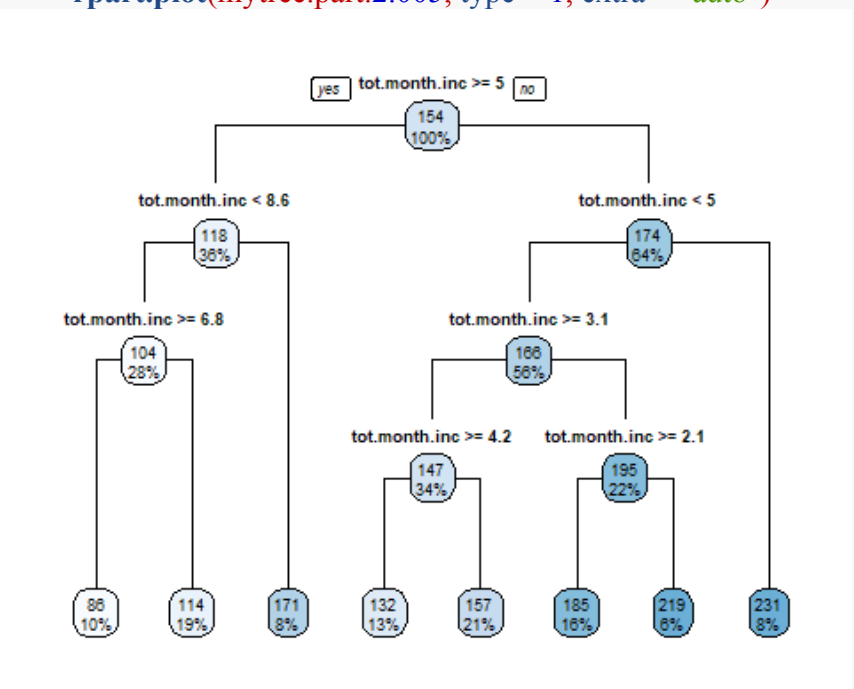
```
mytree.part.2.005<- rpart(loan.value~.,
            data=loan.train,
            method="anova",
            minsplit=2,
            minbucket=1,
            cp=0.005)
mytree.part.2.005
```

```
tree.005<-rpart.plot(mytree.part.2.005, type = 1, extra = "auto")
```



2. Use the optimal cp value from the last step to rebuild the tree. How many leaf nodes are there in this tree?

```
rpart.plot(mytree.part.2.005, type = 1, extra = "auto")
```




3. Choose one leaf node and interpret the numbers related to this leaf node.

- If a loaners total monthly income is greater than or equal to $5,000; less than $8,600 dollars; and greater than or equal to $6,800 dollars; 10% of loaners will have a loan amount of $86,000 dollars.

4. Estimate a multiple linear regression model using the training set with Ln-Orig as your dependent variable and all the rest as independent variables. Cross-validate both your optimal tree from step 2 and the regression model using the testing set. Which model works better? What is your criterion or criteria in making such assessment?

```
lm.model<-lm(loan.value~ credit.score+ first.home+ tot.month.inc+ purchase.p+
               appraise.origin+ median.state.inc+ ratio.loan2home.pp,
             data = loan.train)
lm.pred<-predict(lm.model, newdata = loan.test)
lm.sse<-sum((lm.pred-loan.test$loan.value)^2)
lm.sse
19399450
mtg.pred<-predict(mytree.part.2.005, newdata = loan.test)
mtg.sse<-sum((mtg.pred-loan.test$loan.value)^2)
mtg.sse
13535548
```

- I would choose a decision tree model because it presents a lower SSE of 13535548 compared to the linear model SSE of 19399450. This indicates that the decision tree model will fit this dataset better.

Cleaning Code

```
setwd("C:/Data Mining/Data Sets")
project2<-read.csv("Churn.csv", header=TRUE, sep=",", fileEncoding="UTF-8-BOM")
subset1<-dplyr::select(project2, "gender", "SeniorCitizen", "Partner", "Dependents", "tenure",
"MonthlyCharges", "Contract", "Churn")
is.na(subset1)
na.omit(subset1)
subset1<-subset1%>%dplyr::rename(senior=SeniorCitizen,
 mcharge=MonthlyCharges)
subset1$tenure<-as.numeric(subset1$tenure)
subset1$mcharge<-as.numeric(subset1$mcharge)
subset1$Partner<-as.factor(subset1$Partner)
subset1$Contract=as.factor(subset1$Contract)
subset1$Churn=as.factor(subset1$Churn)
subset1$Dependents<-as.factor(subset1$Dependents)
subset1$gender=as.factor(subset1$gender)
subset1<-subset1 %>% mutate(senior=recode_factor(senior,
 '1'="Senior",
 '0'="N.S"))
subset1<-subset1 %>% mutate(gender=recode_factor(gender,
 'Female'="F",
 'Male'="M"))
subset1<-subset1 %>% mutate(Churn=recode_factor(Churn,
 'Yes'="Yes.churn",
 'No'="No.churn"))
subset1<-subset1 %>% mutate(Contract=recode_factor(Contract,
 'Month-to-month'="M2M",
 'One year'="1 yr or above",
 'Two year'="1 yr or above"))
library(readxl)
mortgagedefaul <- read_excel("C:/Data Mining/Data Sets/mortgagedefaul.xlsx",
    sheet = "MortgageDefaulters")
subset2<-dplyr::select (mortgagedefaul, "Bo_Age", "Credit_score", "First_home",
"Tot_mthly_incm", "pur_prc_amt", "orig_apprd_val_amt", "Median_state_inc", "Ln_Orig",
"Orig_LTV_Ratio_Pct")
subset2<-subset2%>%dplyr::rename(age=Bo_Age,
 credit.score=Credit_score, first.home= First_home, tot.month.inc= Tot_mthly_incm,
purchase.p=pur_prc_amt, appraise.origin=orig_apprd_val_amt,
median.state.inc=Median_state_inc, ratio.loan2home.pp= Orig_LTV_Ratio_Pct,
loan.value=Ln_Orig )
is.na(subset2)
na.omit(subset2)
subset2<-subset2%>%mutate(tot.month.inc=tot.month.inc/1000)
subset2<-subset2%>%mutate(loan.value=loan.value/1000)
```