Week_03_Rui_Peng.R

raypeng

2025-03-29

```
#Step 1: Read in the Data
#Read the data into R
library(rpart) #to use decision tree
library(rpart.plot) #display the decision tree
library(ROCR) #print and see how acurate it is

PATH = "/Users/raypeng/Documents/IS 5213 Data science and big data/HMEQ_Scrubbed"
FILE_NAME = "HMEQ_Scrubbed.csv"

INFILE = paste(PATH, FILE_NAME, sep = "/")

setwd(PATH)
df = read.csv(FILE_NAME)

#List the structure of the data (str)
str(df)
```

```
## 'data.frame':    5960 obs. of  29 variables:
##  $ TARGET_BAD_FLAG    : int  1 1 1 1 0 1 1 1 1 1 ...
##  $ TARGET_LOSS_AMT    : int  641 1109 767 1425 0 335 1841 373 1217 1523 ...
##  $ LOAN               : int  1100 1300 1500 1500 1700 1700 1800 1800 2000 2000 ...
##  $ IMP_MORTDUE        : num  25860 70053 13500 65000 97800 ...
##  $ M_MORTDUE          : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ IMP_VALUE          : num  39025 68400 16700 89000 112000 ...
##  $ M_VALUE            : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ IMP_YOJ            : num  10.5 7 4 7 3 9 5 11 3 16 ...
##  $ M_YOJ              : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ IMP_DEROG          : int  0 0 0 1 0 0 3 0 0 0 ...
##  $ M_DEROG            : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ IMP_DELINQ         : int  0 2 0 1 0 0 2 0 2 0 ...
##  $ M_DELINQ           : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ IMP_CLAGE          : num  94.4 121.8 149.5 174 93.3 ...
##  $ M_CLAGE            : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ IMP_NINQ           : int  1 0 1 1 0 1 1 0 1 0 ...
##  $ M_NINQ             : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ IMP_CLNO           : int  9 14 10 20 14 8 17 8 12 13 ...
##  $ M_CLNO             : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ IMP_DEBTINC        : num  35 35 35 35 35 ...
##  $ M_DEBTINC          : int  1 1 1 1 1 0 1 0 1 1 ...
##  $ FLAG.Job.Mgr       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG.Job.Office    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ FLAG.Job.Other     : int  1 1 1 0 0 1 1 1 1 0 ...
```

```
##  $ FLAG.Job.ProfExe   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG.Job.Sales     : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ FLAG.Job.Self      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG.Reason.DebtCon: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG.Reason.HomeImp: int  1 1 1 0 1 1 1 1 1 1 ...
```

```r
#Execute a summary of the data
summary(df)
```

```
##  TARGET_BAD_FLAG  TARGET_LOSS_AMT      LOAN        IMP_MORTDUE
##  Min.   :0.0000   Min.   :    0   Min.   : 1100   Min.   :  2063
##  1st Qu.:0.0000   1st Qu.:    0   1st Qu.:11100   1st Qu.: 48139
##  Median :0.0000   Median :    0   Median :16300   Median : 65000
##  Mean   :0.1995   Mean   : 2676   Mean   :18608   Mean   : 72999
##  3rd Qu.:0.0000   3rd Qu.:    0   3rd Qu.:23300   3rd Qu.: 88200
##  Max.   :1.0000   Max.   :78987   Max.   :89900   Max.   :399550
##    M_MORTDUE         IMP_VALUE         M_VALUE           IMP_YOJ
##  Min.   :0.00000   Min.   :  8000   Min.   :0.00000   Min.   : 0.000
##  1st Qu.:0.00000   1st Qu.: 66490   1st Qu.:0.00000   1st Qu.: 3.000
##  Median :0.00000   Median : 89000   Median :0.00000   Median : 7.000
##  Mean   :0.08691   Mean   :101536   Mean   :0.01879   Mean   : 8.756
##  3rd Qu.:0.00000   3rd Qu.:119005   3rd Qu.:0.00000   3rd Qu.:12.000
##  Max.   :1.00000   Max.   :855909   Max.   :1.00000   Max.   :41.000
##      M_YOJ           IMP_DEROG          M_DEROG         IMP_DELINQ
##  Min.   :0.00000   Min.   : 0.0000   Min.   :0.0000   Min.   : 0.000
##  1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.000
##  Median :0.00000   Median : 0.0000   Median :0.0000   Median : 0.000
##  Mean   :0.08641   Mean   : 0.3431   Mean   :0.1188   Mean   : 0.503
##  3rd Qu.:0.00000   3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.: 1.000
##  Max.   :1.00000   Max.   :10.0000   Max.   :1.0000   Max.   :15.000
##     M_DELINQ          IMP_CLAGE         M_CLAGE           IMP_NINQ
##  Min.   :0.00000   Min.   :   0.0   Min.   :0.00000   Min.   : 0.00
##  1st Qu.:0.00000   1st Qu.: 117.4   1st Qu.:0.00000   1st Qu.: 0.00
##  Median :0.00000   Median : 174.0   Median :0.00000   Median : 1.00
##  Mean   :0.09732   Mean   : 179.5   Mean   :0.05168   Mean   : 1.17
##  3rd Qu.:0.00000   3rd Qu.: 227.1   3rd Qu.:0.00000   3rd Qu.: 2.00
##  Max.   :1.00000   Max.   :1168.2   Max.   :1.00000   Max.   :17.00
##      M_NINQ           IMP_CLNO         M_CLNO          IMP_DEBTINC
##  Min.   :0.00000   Min.   : 0.00   Min.   :0.00000   Min.   :  0.5245
##  1st Qu.:0.00000   1st Qu.:15.00   1st Qu.:0.00000   1st Qu.: 30.7632
##  Median :0.00000   Median :20.00   Median :0.00000   Median : 35.0000
##  Mean   :0.08557   Mean   :21.25   Mean   :0.03725   Mean   : 34.0393
##  3rd Qu.:0.00000   3rd Qu.:26.00   3rd Qu.:0.00000   3rd Qu.: 37.9499
##  Max.   :1.00000   Max.   :71.00   Max.   :1.00000   Max.   :203.3122
##    M_DEBTINC       FLAG.Job.Mgr     FLAG.Job.Office  FLAG.Job.Other
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.2126   Mean   :0.1287   Mean   :0.1591   Mean   :0.4007
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  FLAG.Job.ProfExe FLAG.Job.Sales    FLAG.Job.Self     FLAG.Reason.DebtCon
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
```

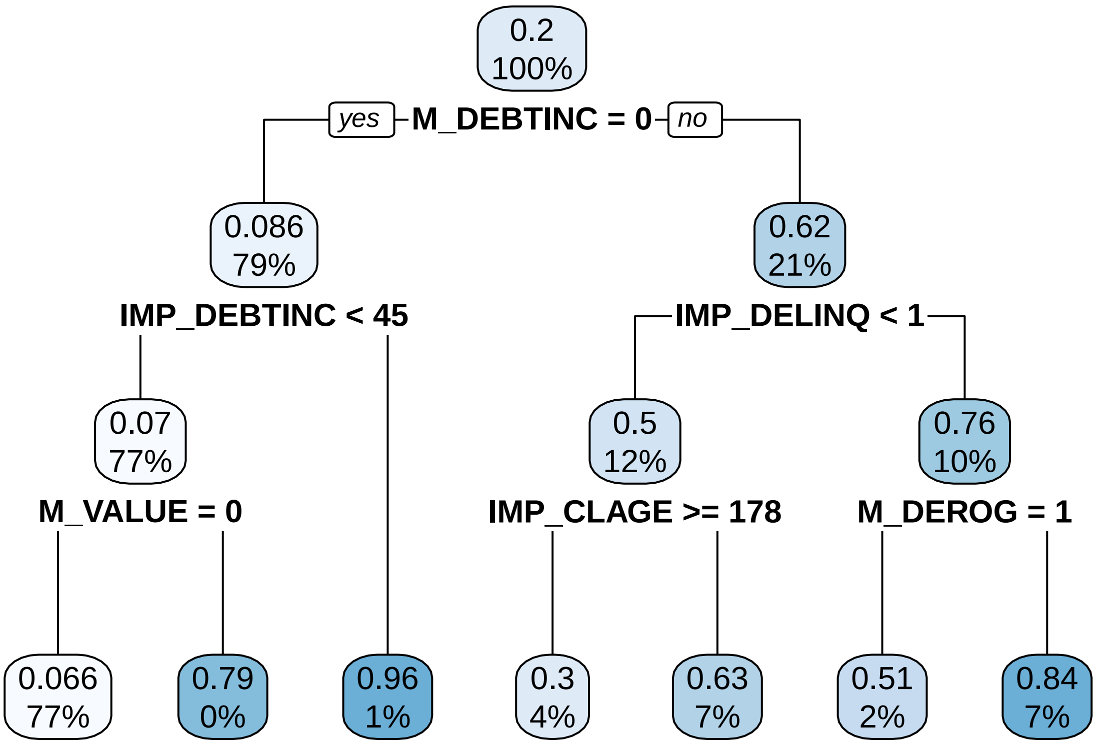
```
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :1.0000
##  Mean   :0.2141   Mean   :0.01829   Mean   :0.03238   Mean   :0.6591
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##  FLAG.Reason.HomeImp
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2987
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```r
#Print the first six records
head(df)
```

```
##   TARGET_BAD_FLAG TARGET_LOSS_AMT LOAN IMP_MORTDUE M_MORTDUE IMP_VALUE M_VALUE
## 1               1             641 1100       25860         0     39025       0
## 2               1            1109 1300       70053         0     68400       0
## 3               1             767 1500       13500         0     16700       0
## 4               1            1425 1500       65000         1     89000       1
## 5               0               0 1700       97800         0    112000       0
## 6               1             335 1700       30548         0     40320       0
##   IMP_YOJ M_YOJ IMP_DEROG M_DEROG IMP_DELINQ M_DELINQ IMP_CLAGE M_CLAGE
## 1    10.5     0         0       0          0        0  94.36667       0
## 2     7.0     0         0       0          2        0 121.83333       0
## 3     4.0     0         0       0          0        0 149.46667       0
## 4     7.0     1         1       1          1        1 174.00000       1
## 5     3.0     0         0       0          0        0  93.33333       0
## 6     9.0     0         0       0          0        0 101.46600       0
##   IMP_NINQ M_NINQ IMP_CLNO M_CLNO IMP_DEBTINC M_DEBTINC FLAG.Job.Mgr
## 1        1      0        9      0    35.00000         1            0
## 2        0      0       14      0    35.00000         1            0
## 3        1      0       10      0    35.00000         1            0
## 4        1      1       20      1    35.00000         1            0
## 5        0      0       14      0    35.00000         1            0
## 6        1      0        8      0    37.11361         0            0
##   FLAG.Job.Office FLAG.Job.Other FLAG.Job.ProfExe FLAG.Job.Sales FLAG.Job.Self
## 1               0              1                0              0             0
## 2               0              1                0              0             0
## 3               0              1                0              0             0
## 4               0              0                0              0             0
## 5               1              0                0              0             0
## 6               0              1                0              0             0
##   FLAG.Reason.DebtCon FLAG.Reason.HomeImp
## 1                   0                   1
## 2                   0                   1
## 3                   0                   1
## 4                   0                   0
## 5                   0                   1
## 6                   0                   1
```

```r
#Step 2: Classification Decision Tree
#Use the rpart library to predict the variable TARGET_BAD_FLAG
```

```r
df_flag = df

#Do not use TARGET_LOSS_AMT to predict TARGET_BAD_FLAG.
df_flag$TARGET_LOSS_AMT = NULL

#All other parameters such as tree depth are up to you.
tr_set = rpart.control( maxdepth = 3 )

tree_flag = rpart( data = df_flag, TARGET_BAD_FLAG ~ ., control = tr_set )
rpart.plot( tree_flag )
```

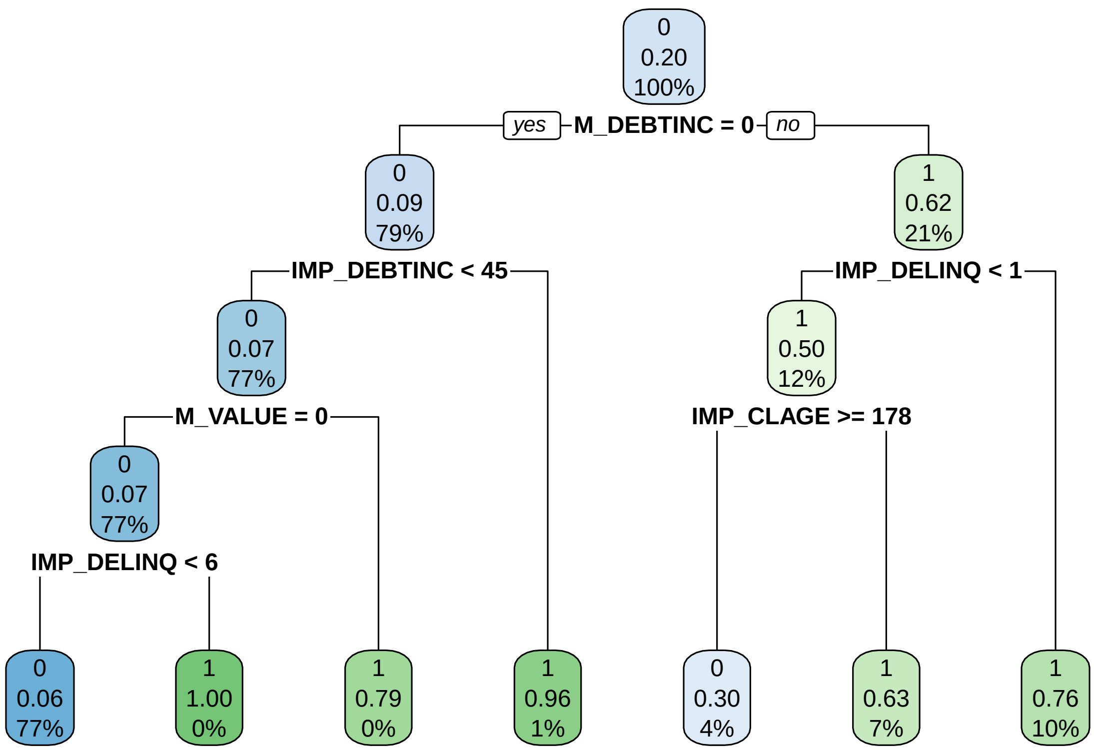



```r
tree_flag$variable.importance
```

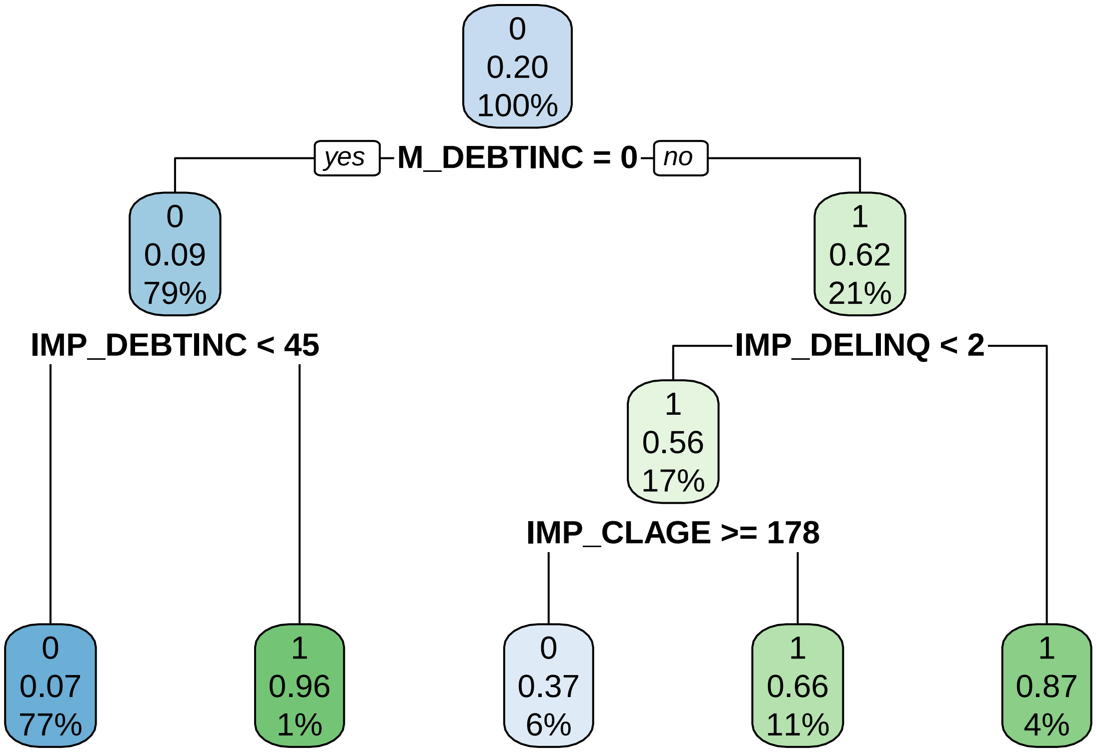
```
##    M_DEBTINC IMP_DEBTINC   IMP_DELINQ      M_VALUE    IMP_CLAGE      M_DEROG
## 285.0105051  64.2695360  28.1876612  25.6672429  18.0381475  15.6708280
##        LOAN   IMP_DEROG     M_DELINQ       M_NINQ       M_CLNO      M_CLAGE
##  12.8228373  11.2507816  10.3565554   8.5221495   6.9916002   4.8633155
##   IMP_VALUE      IMP_YOJ     IMP_CLNO  IMP_MORTDUE        M_YOJ
##   4.2755103   2.1618753   1.4187307    0.8107033    0.2515508
```

```r
#Develop two decision trees, one using Gini and the other using Entropy
tr_set = rpart.control( maxdepth = 10 )
t1G = rpart( data = df_flag, TARGET_BAD_FLAG ~.,
             control = tr_set, method = "class", parms = list(split = 'gini'))
```

```r
t1E = rpart( data = df_flag, TARGET_BAD_FLAG ~.,
             control = tr_set, method = "class", parms = list(split = 'information'))

#Plot both decision trees
rpart.plot( t1G )
```




```r
rpart.plot( t1E )
```




```r
#List the important variables for both trees
t1G$variable.importance
```

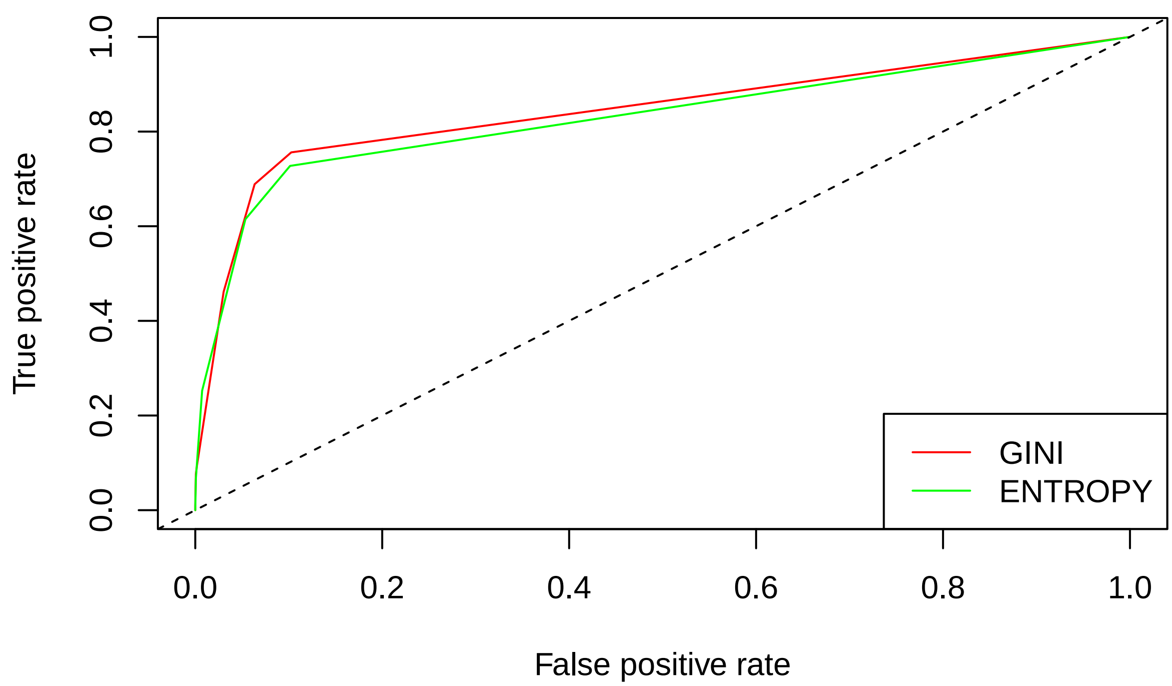
```
##    M_DEBTINC IMP_DEBTINC  IMP_DELINQ     M_VALUE   IMP_CLAGE        LOAN
##   570.021010  128.539072   77.371518   51.334486   36.076295   25.645675
##    IMP_DEROG     M_DEROG   IMP_VALUE    M_DELINQ      M_NINQ     IMP_YOJ
##    22.501563    9.540586    8.551021    7.632469    6.311465    4.323751
##       M_CLNO    IMP_CLNO IMP_MORTDUE
##     4.256569    2.837461    1.621407
```

```r
t1E$variable.importance
```

```
##    M_DEBTINC IMP_DEBTINC  IMP_DELINQ   IMP_CLAGE        LOAN     M_VALUE
##   762.591210  188.922871   68.152477   40.125205   34.053718   30.094365
##    IMP_DEROG   IMP_VALUE     IMP_YOJ    IMP_CLNO IMP_MORTDUE
##    12.037746   10.263083    3.436136    3.075170    1.219274
```

```r
#Create a ROC curve for both trees
pG = predict ( t1G, df )
pG2 = prediction( pG[,2], df$TARGET_BAD_FLAG )
pG3 = performance( pG2, "tpr", "fpr")

pE = predict ( t1E, df )
pE2 = prediction( pE[,2], df$TARGET_BAD_FLAG )
```

```
pE3 = performance( pE2, "tpr", "fpr")

plot( pG3, col = "red" )
plot( pE3, col = "green", add = TRUE )
abline( 0,1,lty=2 )
legend("bottomright", c("GINI","ENTROPY"),
       col = c("red", "green"), bty = "y", lty = 1)
```




```
aucG = performance( pG2, "auc" )@y.values
aucE = performance( pE2, "auc" )@y.values

print(aucG)
```

```
## [[1]]
## [1] 0.8433084
```

```
print(aucE)
```

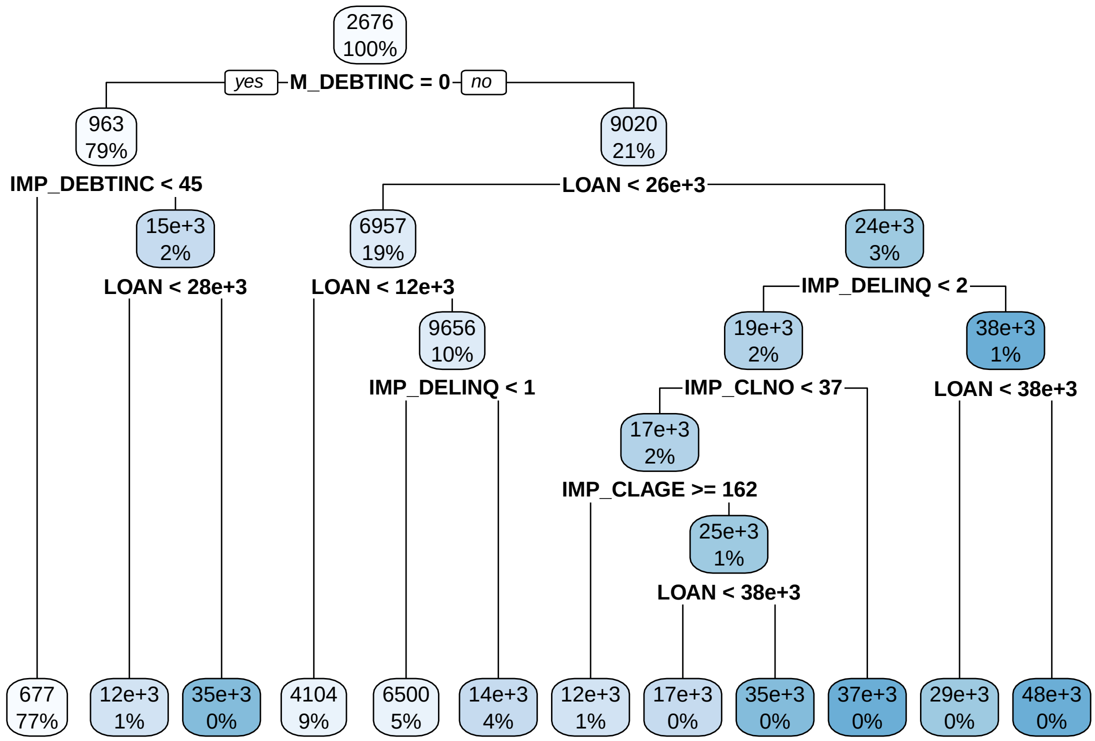
```
## [[1]]
## [1] 0.8293732
```

```
#Write a brief summary of the decision trees discussing whether or not they make sense.
#Summary: both of the gini and entropy trees make sense.
```

```r
#Because the they are both above the random guess line (black dash line).

#Which tree would you recommend using? What type of person will default on a loan?
#I recommend using the red Gini one because it has a larger area under the curve.
#Gini one has the area of 0.8433084 which is larger than 0.8293732 of the entropy one.
#So according to the Gini decision tree, those persons tend to default on a loan:
#Debt income ratio more or equal to 45 (0.96 possibility).
#Who have been late on bills. Who have a credit line age shorter than 178 months (0.63 possibility)

#Step 3: Regression Decision Tree
#Use the rpart library to predict the variable TARGET_LOSS_AMT
df_amt = df

#Do not use TARGET_BAD_FLAG to predict TARGET_LOSS_AMT.
df_amt$TARGET_BAD_FLAG = NULL
mean( df_amt$TARGET_LOSS_AMT )
```

```
## [1] 2676.163
```

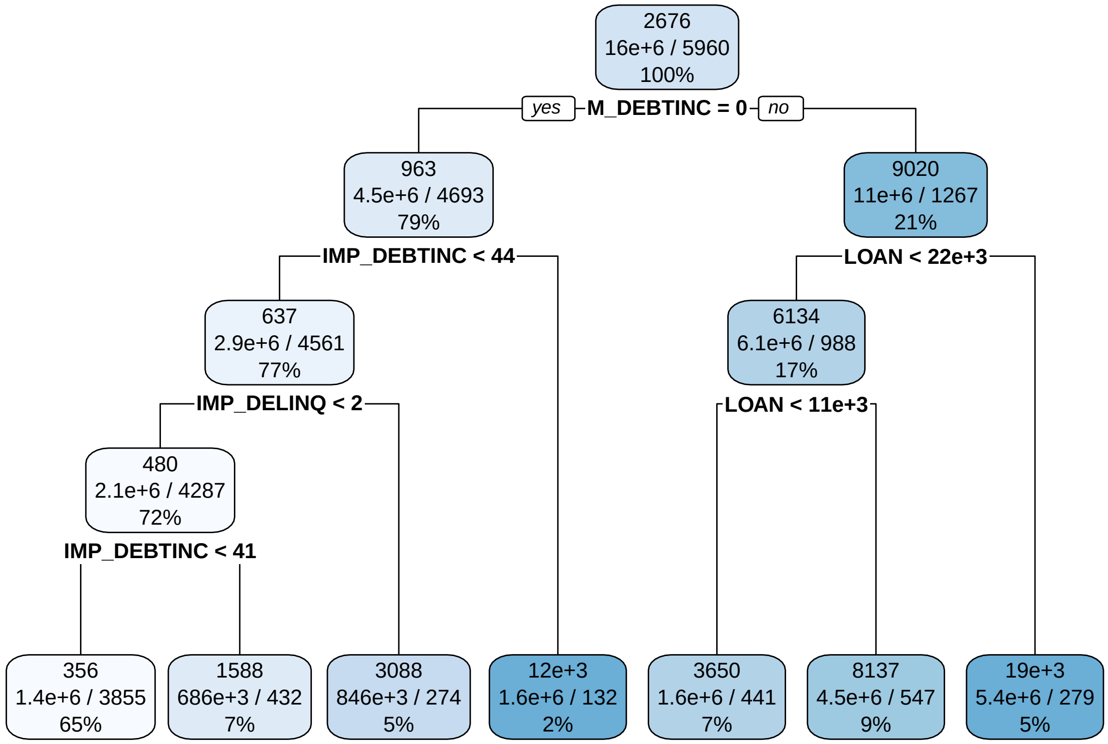
```r
#All other parameters such as tree depth are up to you.
tr_set = rpart.control( maxdepth = 10 )

#Develop two decision trees, one using anova and the other using poisson
t1a = rpart(data = df_amt, TARGET_LOSS_AMT ~ .,
            control = tr_set, method = "anova")

#Plot both decision trees
rpart.plot( t1a )
```




```r
#List the important variables for both trees
t1a$variable.importance
```

```
##           M_DEBTINC                LOAN         IMP_DEBTINC          IMP_DELINQ
##         64758513590         64443856477         19307937442         18468415581
##           IMP_VALUE            IMP_CLNO         IMP_MORTDUE           IMP_CLAGE
##          9985413830          8640006256          7345104792          5561821234
##             M_VALUE           IMP_DEROG FLAG.Reason.HomeImp FLAG.Reason.DebtCon
##          3812596217          3423606021          2487025698          2376139202
##             M_DEROG            M_DELINQ              M_NINQ             IMP_YOJ
##          1695086247          1384320435          1101806061           803802835
##               M_YOJ      FLAG.Job.Other           M_MORTDUE       FLAG.Job.Self
##           727900700           569633461           363950350           269034105
```

```r
#Calculate the Root Mean Square Error (RMSE) for both trees
p1a = predict( t1a, df )
RMSE1a = sqrt( mean( ( df$TARGET_LOSS_AMT - p1a )^2 ) )


t1p = rpart( data = df_amt, TARGET_LOSS_AMT ~ .,
             control = tr_set, method = "poisson" )
rpart.plot( t1p )
```




```
t1p$variable.importance
```

```
##           M_DEBTINC         IMP_DEBTINC                LOAN          IMP_DELINQ
##         18534649.01          6636788.15          5093017.45          1989199.88
##           IMP_VALUE             M_VALUE         IMP_MORTDUE           IMP_DEROG
##           765775.84           731438.40           390250.40           292575.36
## FLAG.Reason.HomeImp FLAG.Reason.DebtCon            IMP_CLNO             IMP_YOJ
##           214334.43           197111.13            82289.11            24796.57
##       FLAG.Job.Self
##            12398.29
```

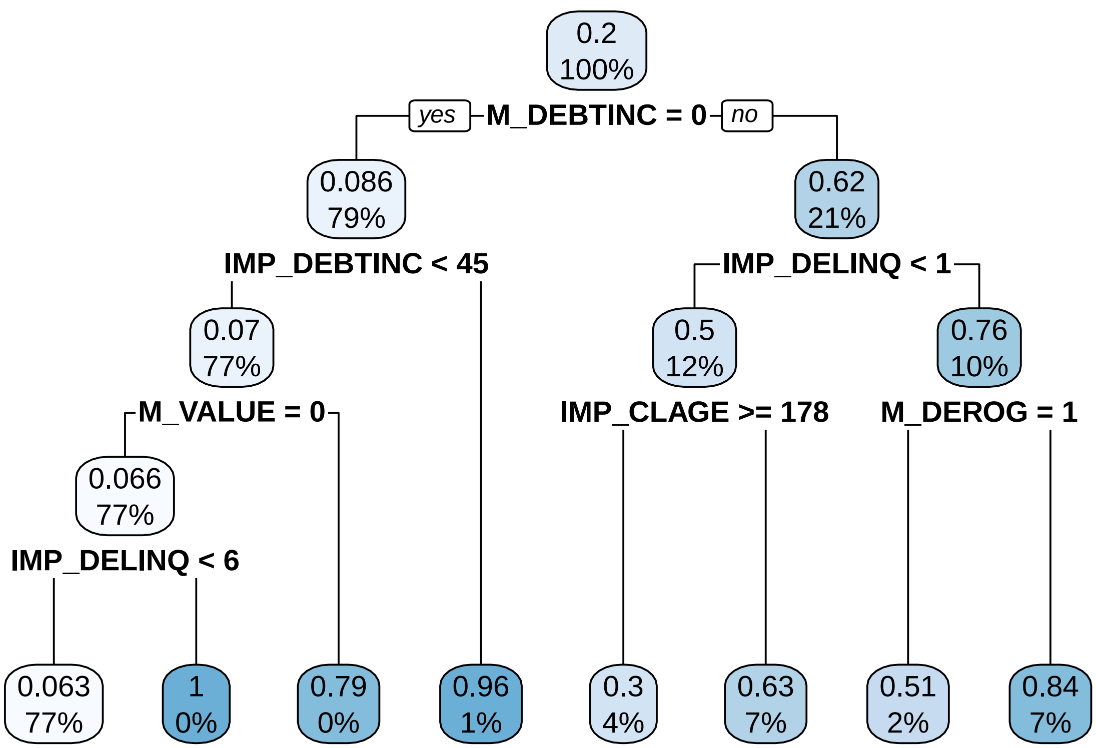
```
p1p = predict ( t1p, df )
RMSE1p = sqrt( mean( ( df$TARGET_LOSS_AMT - p1p )^2 ) )

print( RMSE1a )
```

```
## [1] 4848.417
```

```
print( RMSE1p )
```

```
## [1] 5558.973
```

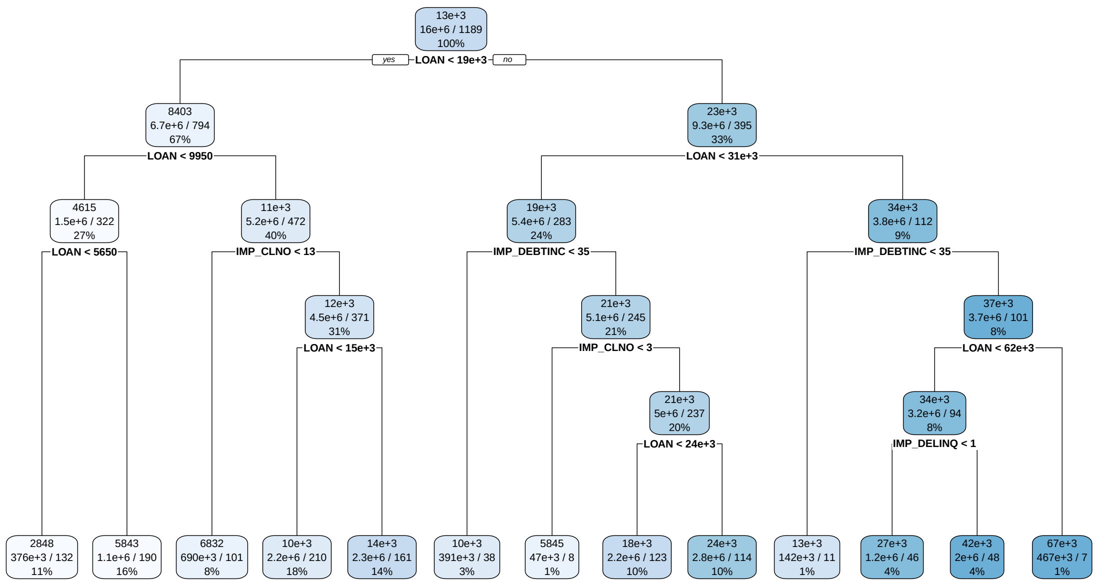
```r
#Write a brief summary: whether or not they make sense. Which tree would you recommend using?
#The models make sense and I would recommend Anova tree
#because it has less prediction error (4848.417) compared to Poisson tree (5558.973).

#What factors dictate a large loss of money?
#According to the anova chart, there are two main causing big loss of money:
#Number one reason: Big amount of loan.
#Number two reason: Credit lines. The more credit lines the persons have, the larger amount of money it

#Step 4: Probability / Severity Model Decision Tree (Push Yourself!)
#Use the rpart library to predict the variable TARGET_BAD_FLAG
df_flag = df
df_flag$TARGET_LOSS_AMT = NULL

t2_f = rpart( data = df_flag, TARGET_BAD_FLAG ~ ., control = tr_set )

#Plot both decision trees
rpart.plot( t2_f )
```




```r
p2_f = predict ( t2_f, df )

#Use the rpart library to predict the variable TARGET_LOSS_AMT using only records where TARGET_BAD_FLAG
df_amt_2 = subset( df, TARGET_BAD_FLAG == 1)
df_amt_2$TARGET_BAD_FLAG = NULL
head(df_amt_2)
```

```
##   TARGET_LOSS_AMT LOAN IMP_MORTDUE M_MORTDUE IMP_VALUE M_VALUE IMP_YOJ M_YOJ
## 1             641 1100       25860         0     39025       0    10.5     0
## 2            1109 1300       70053         0     68400       0     7.0     0
## 3             767 1500       13500         0     16700       0     4.0     0
## 4            1425 1500       65000         1     89000       1     7.0     1
## 6             335 1700       30548         0     40320       0     9.0     0
## 7            1841 1800       48649         0     57037       0     5.0     0
##   IMP_DEROG M_DEROG IMP_DELINQ M_DELINQ IMP_CLAGE M_CLAGE IMP_NINQ M_NINQ
## 1         0       0          0        0  94.36667       0        1      0
## 2         0       0          2        0 121.83333       0        0      0
## 3         0       0          0        0 149.46667       0        1      0
## 4         1       1          1        1 174.00000       1        1      1
## 6         0       0          0        0 101.46600       0        1      0
## 7         3       0          2        0  77.10000       0        1      0
##   IMP_CLNO M_CLNO IMP_DEBTINC M_DEBTINC FLAG.Job.Mgr FLAG.Job.Office
## 1        9      0    35.00000         1            0               0
## 2       14      0    35.00000         1            0               0
## 3       10      0    35.00000         1            0               0
## 4       20      1    35.00000         1            0               0
## 6        8      0    37.11361         0            0               0
## 7       17      0    35.00000         1            0               0
##   FLAG.Job.Other FLAG.Job.ProfExe FLAG.Job.Sales FLAG.Job.Self
## 1              1                0              0             0
## 2              1                0              0             0
## 3              1                0              0             0
## 4              0                0              0             0
## 6              1                0              0             0
## 7              1                0              0             0
##   FLAG.Reason.DebtCon FLAG.Reason.HomeImp
## 1                   0                   1
## 2                   0                   1
## 3                   0                   1
## 4                   0                   0
## 6                   0                   1
## 7                   0                   1
```

```
t2_a = rpart( data = df_amt_2, TARGET_LOSS_AMT ~ .,
              control = tr_set, method = "poisson" )
rpart.plot(t2_a)
```




```r
p2_a = predict ( t2_a, df )
head( p2_f )
```

```
##          1          2          3          4          5          6
## 0.63084112 0.83710407 0.63084112 0.50769231 0.63084112 0.06344345
```

```r
head( p2_a )
```

```
##        1        2        3        4        5        6
## 2848.089 2848.089 2848.089 2848.089 2848.089 2848.089
```

```r
#List the important variables for both trees
t2_f$variable.importance
```

```
##    M_DEBTINC IMP_DEBTINC   IMP_DELINQ      M_VALUE   IMP_CLAGE      M_DEROG
## 285.0105051  64.2695360   38.6857592   25.6672429  18.0381475   15.6708280
##         LOAN   IMP_DEROG     M_DELINQ       M_NINQ      M_CLNO      M_CLAGE
##  12.8228373  11.2507816   10.3565554    8.5221495   6.9916002    4.8633155
##    IMP_VALUE     IMP_YOJ     IMP_CLNO IMP_MORTDUE       M_YOJ
##    4.2755103   2.1618753    1.4187307    0.8107033   0.2515508
```

```r
t2_a$variable.importance
```

```
##                LOAN           IMP_VALUE         IMP_MORTDUE         IMP_DEBTINC
##          6409665.00          1481448.38          1081934.14           574282.90
##            IMP_CLNO FLAG.Reason.HomeImp          IMP_DELINQ FLAG.Reason.DebtCon
##           446748.28           229285.80           223669.68           188922.21
##       FLAG.Job.Self           IMP_CLAGE            IMP_NINQ           IMP_DEROG
##           147185.77            51185.99            48213.49            45544.27
##             IMP_YOJ             M_VALUE      FLAG.Job.Other
##            38733.92            12118.28             7457.40
```

```r
#Using your models, predict the probability of default and the loss given default.
#Multiply the two values together for each record.
p2 = p2_f * p2_a
head( p2 )
```

```
##         1         2         3         4         5         6
## 1796.6918 2384.1472 1796.6918 1445.9530 1796.6918  180.6926
```

```r
#Calculate the RMSE value for the Probability / Severity model.
RMSE2 = sqrt( mean( (df$TARGET_LOSS_AMT - p2 )^2 ))
print(RMSE2)
```

```
## [1] 4830.517
```

```r
#Comment on how this model compares to using the model from Step 3. Which one would your recommend usin
#This one is better than the model from Step 3 because this one has a smaller RMSE of 4830.517
#While in step 3, the RMSE was 4848 and 5559.
```