ROBERT H. SMITH
SCHOOL OF BUSINESS

BUDT733 - Homework 6
Quantitative analysis of credit (contd.)

In this Assignment we will finish our analysis of the Credit data. We use the same data, and the goal remains the same: to create a model that a bank or another financial institution can use to classify a new credit request as accept/not accept (CREDIT_EXTENDED should be excluded from the analysis).

Data Preparation

1) Open the file credit3.xlsx. Create the outcome variable (PROFITABLE=1 if NPV>0, =0 otherwise), create factors for CHK_ACCT, SAV_ACCT, HISTORY, JOB and TYPE variables. Split the data using the sample function; 70% as training data and 30% as test data; setting the seed to 12345. (Do not use NPV as a predictor)

```
df <- read.csv("credit3.csv")
df$CREDIT_EXTENDED <- NULL
df$PROFITABLE <- ifelse(df$NPV > 0, 1, 0)
df$NPV <- NULL
df$OBS. <- NULL
df$CHK_ACCT <- factor(df$CHK_ACCT)
df$SAV_ACCT <- factor(df$SAV_ACCT)
df$HISTORY <- factor(df$HISTORY)
df$JOB <- factor(df$JOB)
df$TYPE <- factor(df$TYPE)
df$PROFITABLE <- factor(df$PROFITABLE)
df$AMOUNT_REQUESTED <- as.numeric(df$AMOUNT_REQUESTED)
#
set.seed(12345)
split <- sample(nrow(df), 0.7 * nrow(df))
train <- data.frame(df[split,])
test <- data.frame(df[-split,])
```

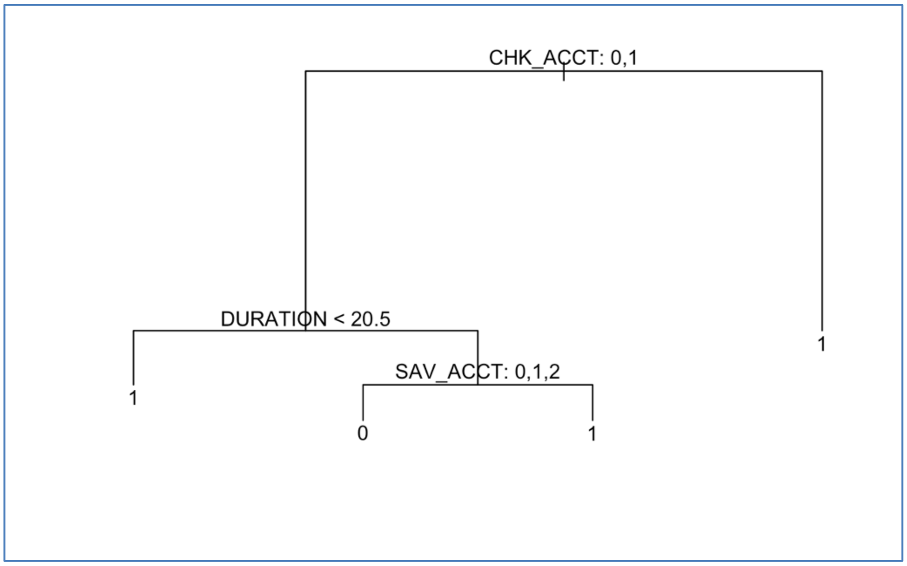
Predicting profitable accounts with Classification Trees

2) Run the Classification Tree algorithm using the data, with the PROFITABLE as the output variable. Set the seed to 123 and then use K-fold cross-validation (with K = 10) to prune back the tree. Attach the classification confusion matrix for the test data as **Exhibit 1** and a figure of the pruned tree as **Exhibit 2**.

Exhibit 1

```
pred.credit  0   1
       0  44  29
       1   45 182
```

Exhibit 2



3) How many decision nodes are in the full tree (using R default values)? 13

How many decision nodes are in the pruned tree? **4**

Which model (the full or pruned tree) gives you better accuracy?
Both the trees have more or less the same accuracy
Full Tree [1] 0.7533333
Pruned Tree [1] 0.7

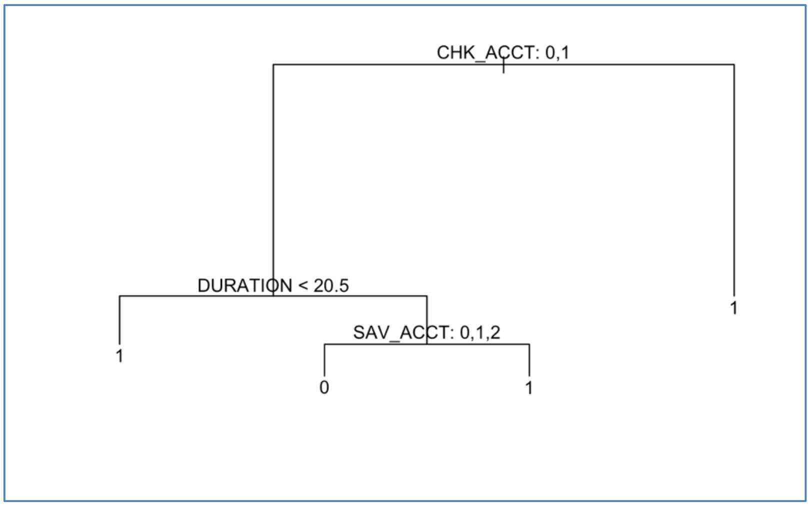
4) How would the tree classify our student from the previous HW (the student is 27 years old, domestic, has $100 in her checking account but no savings account. The applicant has 1 existing credits, and a credit duration of 12 months, and the credit was paid back duly. The applicant has been renting her current place for less than 12 months, does not own any real estate, just started graduate school (the present employment variable is set to 1 and nature of job to 2). The applicant has no dependents and no guarantor. The applicant wants to buy a used car and has requested $4,500 in credit, and therefore the Installment rate is quite high or 2.5%, however the applicant does not have other installment plan credits. Finally, the applicant has a phone in her name)

Profitable / Not Profitable (pick one) **Profitable**

What is the predicted probability that the account is profitable? **0.6570048**

5) Find the best pruned tree with only 4 terminal nodes. Describe the rule in words (in English)

```
Classification tree:
snip.tree(tree = creditTree, nodes = c(3L, 11L, 4L))
Variables actually used in tree construction:
[1] "CHK_ACCT" "DURATION" "SAV_ACCT"
Number of terminal nodes:  4
Residual mean deviance:  1.056 = 735.1 / 696
Misclassification error rate: 0.25 = 175 / 700
```

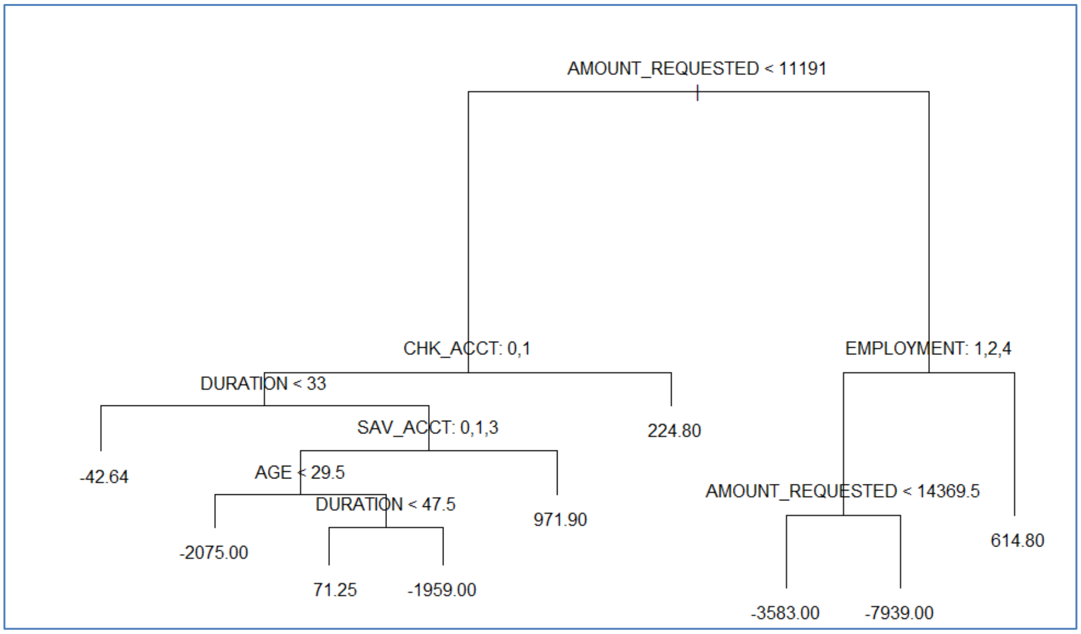


Loan is given out if CHK_ACCT is not 0,1.
Loan is given out if CHK_ACCT is 0,1 and DURATION is less than 20.5
Loan is given out if CHK_ACCT is 0,1 and DURATION is greater than 20.5 and SAV_ACCT is not 0,1,2

Predicting profit with Regression Trees

6) Reset the seed to 123 and run a Regression Tree Algorithm to predict the NPV of each applicant. Use a pruned tree to score the data samples. Attach the pruned tree as **Exhibit 3**.

Exhibit 3



7) In the output for the test sample, the prediction for each node corresponds to the average NPV of all training records in that end-node. Therefore, based on the training data we would extend credit to all requests with a positive predicted NPV.

Score the test data (i.e. compute predicted NPV), create a table that summarizes the number of records from the test data in each end node (each end-node has a distinct prediction value), and the total actual NPV of the test records in these nodes. Attached the table as **Exhibit 4**.

Exhibit 4

		dfPruneTree.Freq
-6985.2	3488	1
-4997.8	-5406	1
-3434.869565	-1134	2
-2975.923077	-4881	2
-1959	-14134	7
-1010.090909	-1821	4
-476.8571429	937	1
-42.64	-14736	126
71.25	204	7
224.7679739	33297	146
971.9	-4844	3

Based on your table and the predicted NPV values, how many customers in the test sample would you extend credit to?

156

What would be the average profit per customer (that you extend credit to)?

183.69

What is the overall profit for all customers you extend credit to in the test sample?

28657

How do these values compare with extending credit to everyone?
Profit from the customers is 38074
Extending credit to all the customers would provide a loss of -9030

8) Compare the pruned classification tree to the pruned regression tree (Exhibits 2 and 3). These two trees are an indicator of what are some of the more important variables when classifying a profitable account and predicting the profit of an account. In what way are these trees similar/dissimilar? Briefly discuss.

The end nodes for the pruned classification tree show 1 and 0 which correspond to whether it is profitable or not-profitable to extend credit to the customer.
The end nodes for the regression tree show the average NPV values when a credit is extended to a customer or not.
Also, the decision nodes for classification tree are different from that of regression tree.

Selecting the "right" customers with multiple linear regression

9) Build a linear multiple regression model to predict NPV. Attach **Exhibit 5** that neatly shows the variables and their corresponding coefficients. Use the training sample to select a cut-off value, which maximizes the overall profit (if the predicted NPV is above the cut-off the bank will extend credit, if the predicted NPV is below the cut-off value, the bank should decline the credit request).

Exhibit 5

```
Call:
lm(formula = NPV ~ . - PROFITABLE, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-8812.6  -468.6   -38.6   493.7  6983.1 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       876.12757  557.09527   1.573 0.116271    
AGE                -1.95907    4.38018  -0.447 0.654835    
```

```
CHK_ACCT1           0.49980  122.74033   0.004 0.996752    
CHK_ACCT2         319.65204  192.54533   1.660 0.097359 .  
CHK_ACCT3         345.70468  117.39306   2.945 0.003344 ** 
SAV_ACCT1          32.27715  154.36648   0.209 0.834439    
SAV_ACCT2         266.93420  189.20115   1.411 0.158758    
SAV_ACCT3         193.99848  219.83611   0.882 0.377843    
SAV_ACCT4         518.74520  121.27145   4.278 2.17e-05 ***
NUM_CREDITS      -129.78287   95.04635  -1.365 0.172568    
DURATION            3.38489    5.03674   0.672 0.501792    
HISTORY1           30.79689  314.35891   0.098 0.921988    
HISTORY2          310.97347  240.28581   1.294 0.196053    
HISTORY3          784.81425  263.65426   2.977 0.003020 ** 
HISTORY4          634.13686  242.82747   2.611 0.009219 ** 
PRESENT_RESIDENT  -23.67110   44.53018  -0.532 0.595199    
EMPLOYMENT        -18.22134   41.45575  -0.440 0.660415    
JOB1              -84.45531  315.08123  -0.268 0.788749    
JOB2             -155.13204  307.77961  -0.504 0.614403    
JOB3               75.70558  329.14939   0.230 0.818160    
NUM_DEPENDENTS    179.31427  121.69882   1.473 0.141110    
RENT             -307.54922  184.97886  -1.663 0.096861 .  
INSTALL_RATE     -152.72527   44.99147  -3.395 0.000728 ***
GUARANTOR          75.34984  215.91882   0.349 0.727220    
OTHER_INSTALL     -67.39237  120.47041  -0.559 0.576070    
OWN_RES          -160.97494  159.22133  -1.011 0.312378    
TELEPHONE         -40.06193  101.55319  -0.394 0.693344    
FOREIGN          -185.05609  245.23304  -0.755 0.450749    
REAL_ESTATE       -24.62598  108.53466  -0.227 0.820575    
TYPE1            -185.56661  205.19635  -0.904 0.366145    
TYPE2             364.47374  236.69001   1.540 0.124066    
TYPE3            -118.41467  214.03791  -0.553 0.580284    
TYPE4            -110.66245  206.09277  -0.537 0.591479    
TYPE5            -726.79999  268.15758  -2.710 0.006895 ** 
TYPE6            -456.69914  234.07918  -1.951 0.051472 .  
AMOUNT_REQUESTED   -0.21496    0.02345  -9.168  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1130 on 664 degrees of freedom
Multiple R-squared:  0.2884,	Adjusted R-squared:  0.2509 
F-statistic:  7.69 on 35 and 664 DF,  p-value: < 2.2e-16
```

HINT: Sort your training sample by the predicted value. Sum up the actual values for each possible cut-off value. Select the cut-off that results in the maximum sum.

What is your optimal cut-off value?

-450

10) Apply the cut-off value to the test sample. How many customers in the test sample would you extend credit to?

230

What would be the average profit per customer (that you extend credit to)?

142

What is the overall profit for all customers you extend credit to in the test sample?

32828

Bagging, Random Forest and Boosting

11) Use boosting, random forest and bagging to examine if performance of your classifier above can be improved. Discuss improvements (or not) both in terms of accuracy and profitability.

Classification Pruned Tree: 0.74
Bagging: 0.7566667
Random Forest: 0.7433333
Boosting: 0.74
In general bagging reduces the variance of the decision tree, it chooses random subsets of data from the training set with replacement. The average of all predictions is the result.
Random forest is an improvement over bagging, in addition to making random subset of data, it also takes random subsets of features used to predict the model.
Boosting tries to make an improvement on the previous tree, the initial trees are computed for errors, and then the model learns from these errors to predict more accurately in the next tree. Its additive rather than averaged.
Here in this case, there was no such significant improvement in random forest over bagging or boosting over random forest – hence no single model outperforms another, sometimes the result get better, sometimes worse – in this case the best one was found from bagging