Data 630 – Fall 2018

Assignment 3 – Decision Tree Analysis of Credit Risk

Kenneth Lulie

Dr. Gates

University of Maryland University College

October 28, 2018

**Decision Tree Analysis of Credit Risk using the German Credit Dataset**

**Introduction**

One of the most important aspects to a healthy economy is a well-functioning credit market. The ability to efficiently and safely loan capital to people or entities that need it in exchange for interest is one of the backbones of any capitalist system. However, as shown in the 2008 credit crisis, when banks recklessly loan money to high risk borrowers there can be drastic consequences. If enough loans at a single bank go into default where the borrower is unable to repay the loan, the bank may enter bankruptcy, and if enough banks go bankrupt the entire economy may be negatively affected. Therefore, being able to understand the different characteristics of a safe loan versus that of a risky loan is essential to the long term health of any bank and as well as the economy.

There has been a great deal of research to identify the characteristics of a good borrower versus that of a risky borrower (Gonzalez-Garcia). In the United States, potential borrowers are assigned credit scores by credit bureaus which keep track of factors such as the amount of total accounts you own, the average age of those accounts, the ratio of used credit to total potential credit and your history of credit payments. The higher your credit score, the better terms you will receive on potential loans offered by banks. Many banks will rely on this credit score as an easy method to classify a potential borrower and set terms without needing to collect costly data themselves.

The cost of a defaulted loan to a bank can be many times the profit gained from that loan. In 2017, 676,535 properties were filed for foreclosure representing .51% of all housing units in the US down from 2.23% in 2010 (Staff, 2018). 6% of total credit cards loan by balance in 2017 were over 90 days delinquent, down from around 14% in 2010. The effects of the Great Recession is widely accepted to have been in full effect during 2010, while 2017 was a year with a strong economy. These statistics shows how the overall health of the economy affects the rate of loan defaults, and how good years can lead banks to give risky loans. It is important to maintain high lending standards even when the economy is strong to limit risk in the event of an unexpected downturn.

In this analysis, the impact of different factors such as age, history of checking accounts, marital status, size of loan, type of loan, purpose of loan, housing status, and job status will be tested to identify their usefulness to predict loan defaults. The specific aim of this analysis is to use the Decision Tree algorithm to test the relationships of these different factors and build a model that predicts the risk of a default. The model will be able to predict the specific odds for a given loan application to go into default. The model will also show the specific rules used to calculate the odds of defaulting. Additionally, the model will be tuned to identify the best thresholds of probability to minimize false positives, which in this case are loans approved and then defaulted. The data description included with the data set suggested creating a model which penalized false positives as -5 points, while penalizing false negatives as -1 point and as such the model thresholds will be scored and tuned based on this score.

**Analysis and Model Demonstration**

**Subsection: Data Information, Cleaning, and Preprocessing**

The data used in this analysis comes from the UMUC provided list of approved datasets, originally provided by Professor Hofmann, and then altered by Strathclyde University. The dataset is that of 1000 loan applications made in Germany in 1994. The loan applications have been made anonymous with no identifying data remaining with each loan classified as “good” or “bad”.

The analysis of this study was implemented using Rstudio version 3.5.1. There were 1000 loan applications in the dataset provided. There were no missing values in the observations provided. The data as provided contained 21 variables including checking status, duration of loan, credit history, purpose of loan, amount requested for the loan, saving status, length of employment history, loan as percentage of disposable income, personal status indicating household/marital status, other parties indicating if there was a loan guarantor or co-applicant, residence status indicating how long they have lived at their current residence, property magnitude indicating property ownership, age, other payment plans, housing status, number of existing credits at the bank, job type, number of dependents, if they own a telephone, if they are a foreign worker, and if the loan was classified as good or bad.

The data was reviewed for missing values and unique identifiers and no missing values or unique identifiers were found. The data was also reviewed outliers or other possible errors in the data. No outliers or errors in the data were found. The data was also reviewed for data types to ensure that the data was of the correct type. The class variable was changed from a two level factor of “bad” or “good” to an integer with 0 corresponding to “bad” and 1 corresponding to “good”. A summary of the frequency of the categorical variables are presented in (**Table 2)**. There were 300 loan applications were labeled as “bad” and 700 loan applications were labeled as “good”.

Table 1 Distribution of Numeric Variables

|  |  |  |  |
| --- | --- | --- | --- |
| Age | | Duration | |
| Min | 19 | Min | 4 |
| 1st Qu. | 27 | 1st Qu. | 12 |
| Median | 33 | Median | 18 |
| Mean | 35.55 | Mean | 20.9 |
| 3rd Qu. | 42 | 3rd Qu. | 24 |
| Max. | 75 | Max. | 72 |
| Credit Amount | | Resident Since | |
| Min | 250 | Min | 1 |
| 1st Qu. | 1366 | 1st Qu. | 2 |
| Median | 2320 | Median | 3 |
| Mean | 3271 | Mean | 2.845 |
| 3rd Qu. | 3972 | 3rd Qu. | 4 |
| Max. | 18424 | Max. | 4 |

The distribution appears to be skewed towards males, with 690 of the 1000 applications coming from a male. This is likely due to the time period of 1994, as we would expect a more even split today. Only 93 the loans had a guarantor or co-applicant, and many of the applicants had a “critical” credit history. We can see from this that there is a diverse set in quality of loan applicants. A descriptive analysis including mean, median, 1st and 3rd quartiles, min and max of the Duration, Credit Amount, Residence Since, and Age variables are provided in **Table 1**. The boxplot visualization of Credit Amount shows us that the average credit requested is generally under 3,000, but many outliers exist **(Figure 1).** The boxplot data visualization for Age and Credit Duration show that Age is skewed above the median, and that Duration of the Loan does not generally contain outliers (**Figure 2**).

**Subsection: Analysis and Model Methods**

The preprocessed data was used in this analysis with the Decision Tree algorithm for classification to predict if the loan should be classified as bad or good. A simultaneous result is also the logic of how the decision tree is used, showing us the nodes and logic on how each variable affects the decision of ultimately predicting the class variable of good or bad. The Decision Tree algorithm can also show on a scale of 0 to 1 the percent of each variable being predicted as a good or bad credit. This can allow us to fine tune our prediction thresholds to maximize the models score based on penalizing false positives more than false negatives. The information shown in the decision tree is also valuable information on what the most important variables are when it comes to predicting if someone will default on a loan, and how those variables either positively or negatively affect those chances. This additional information can be useful to loan officers who may receive a recommendation from an algorithm and may have discretion on how to use that recommendation.

The Decision Tree algorithm is a supervised classification or regression machine learning method which builds a series of nodes based on breaking the dataset down into smaller and smaller subsets based on logical tests (Han, 2011). Each observation starts at the root node and gets sent from one node to the next based upon tests such as the value of a factor or if the value of a variable is below or above a certain threshold for a numeric value. The data starts in a single node, labeled the root node, at the top of the tree which has no incoming edges, and only outgoing edges. Next, data is passed to the Internal Nodes, each of which has exactly one incoming edge and can have two or more outgoing edges. Finally, the data arrives at a leaf or terminal node which has no more outgoing edges.

One of the strengths of a decision tree is that it can also be used to calculate probabilities of different classifications based upon the proportion of outcomes in a terminal node. This gives an indication as to how confident the model is on each prediction of classification made. This makes it possible to fine tune the model into lower or higher thresholds, an application that can be used to make more intelligent choices in areas such as credit loans.

To measure the effectiveness of the Decision Tree model a confusion matrix will be used to review the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). From these numbers Error Rate (FP + FN)/Total Number of Predictions), Accuracy (TP + TN/Total Number of Predictions), Sensitivity (TP/ Total Actual Positives), and Specificity (TN/Total Actual Negatives), Precision (TP/TP + FP), False Positive Rate (FN/TN+ FN) can be calculated and used to measure the model. However, the main measures of the model will be the ROC Curve, and the score of false positives and false negatives as previously described. ROC is an abbreviation for “Receiver Operating Characteristic” and is created by plotting the True Positive Rate against the False Positive Rate at various threshold settings. A score is also generated from 0 to 1 showing the percentage of area in the graph under the curve. A 1 indicates a perfect score, with a 0 indicating the inverse. However, it is worth noting that if a model actually had a AUC score of 0, the results could simply be flipped to get a perfect score.

This analysis reviewed the results of one decision tree model at three different thresholds using the ctree Decision Tree algorithm in R with the default settings. The data was randomly sampled with 70% going to a training dataset, and the remaining 30% going to the testing dataset. The model was trained on the training set, and then used to predict good or bad classifications for the test dataset. The model was evaluated at three different thresholds for the probability of being a good creditor, .5, .35, and .75. The model was also evaluated with a ROC Curve. Each threshold was measured using a Confusion Matrix, with the Error Rate, Accuracy, Sensitivity, Specificity, Precision, and False Positive Rate calculated. Each threshold also had a score calculated to measure the amount of false positives and false negatives where false positives were scored as -5 points and false negatives were scored as -1 points. An ideal model would have a score of 0, and the goal is to identify the threshold with the score closest to 0.

**Results**

The decision tree model produced a decision tree with 8 terminal nodes and 6 internal nodes and 1 root node. A visualization of the decision tree is provided in Figure 3. The written rules the decision tree produced are shown in Appendix 1. The Decision Tree model had an AUC score of .7498. A ROC Curve was generated and indicated a relatively usual curve indicating the accuracy of the model will decrease relatively the same when changing the threshold up or down showing that the model will permit minimizing false negatives at the expense of increasing false positives or vice versa **(Figure 4)**. The variables used to determine the splits were checking status, duration, credit history, other payment plans, and credit amount indicating that these were the most important variables to differentiate from good or bad loans. Reviewing the rules, we can see that the root node splits on checking status, with values of <0, indicating a negative balance, or between 0 and 200 leads to the right side of the tree with terminal nodes indicating a lower chance of repaying the loans than the population split of 70% good and 30% bad, with status of greater than 200 or no checking leading to the left side which has generally higher proportions of good borrowers. Additionally, credit history is also used on the left and right sides with values of critical/other existing leading to terminal nodes with lower proportions of good creditors as well. Duration is used in an internal node where it is split based on lower than or equal to 21 versus greater than 21, with the higher duration loans going to a terminal node with much lower than the population percentage of good borrowers. Credit amount is also used in an internal node split on less than or equal to 6742 or greater than 6742, with loans greater than 6742 leading to a lower percentage of the loan being good.

The decision tree model produced had an accuracy of 71.85% when run against the test data with a threshold of .5 **(Table 3)**. The precision rate was 75.83% indicating that about three out of every four positives predicted were actually positive, with the one remaining being a false positive. The false positive rate was 47.22% indicating that about 1 in every 2 loans that were actually bad were predicted as good. The false negative rate was 17.5% indicating that less than 1 in 5 of loans actually good were predicted as false. When scored for -5 points for each false positive and -1 point for each false negative the model had -289 points as it had 51 false positives, and 34 false negatives. This model had the highest overall accuracy of the three thresholds tested.

The decision tree model had an accuracy of 63.25% when run against the test data with a threshold of .75, indicating a stricter test that only would classify observations as positive if the model thought it was at least 75% or higher of being positive **(Table 4)**. While the stricter test caused the accuracy to be lowered, the false positive rate went from 47.22% in the first model to 13.89%. The false negative rate went from 17.5% to about 50%. This indicates the change in threshold made the model improve in accuracy when it predicts for a positive outcome, or a good credit classification, but much worse in accuracy on the loans it predicts to be negative. When scored, it had a score of -171 with 15 false positives and 96 false negatives. This was an improvement in score of 108 points and was the best score of all three thresholds tested.

The decision tree model had an accuracy of 66.56% when run against the test data with a threshold of .35, indicating a weaker test that would classify observations as positive if the model thought the probability was 35% or higher of being positive **(Table 5)**. This weaker test had a false positive rate of 91.67% indicating that a bad credit had a 91.67% likelihood of being classified as good. The lowering of the threshold combined with the population percentage of 70% good to 30% bad resulted in a model that is almost identical to simply predicting every observation as good, with only 11 out of 302 in the test data being predicted as false. When scored, it had a score of -497 points with 99 false positives and 2 false negatives.

Of the three models, the model with the best score was the threshold of .75. As might be expected, due to the score weighing of false positives more strongly than false negatives, a strategy of a high threshold earns the best score as it will let through only the observations it is most confident at, in exchange for many missed opportunities. To translate this to business terms, higher thresholds are best for business applications where a false positive is expensive such as this subject case of loan applications, even when you factor in missing out on the profit of loans that would have been good which were rejected. However, if the application was one where false positives weren’t penalized heavily such as advertising, a different threshold strategy could be best.

**Conclusion**

This analysis showed that the decision tree algorithm combined with the variables provided could be used to build a model to classify loans into good or bad buckets, and identify the most important variables and how they affected the prediction. Additionally, it showed that a strategy of adjusting the threshold of used to predict between classes could be used to reduce the rate of false positives at the expense of increased false negatives. This strategy could be useful in applications which value minimizing false negatives or false positives even at the expense of lower overall accuracy such as some medical tests or the used dataset of credit loans.

Future studies could improve on the limitations of this analysis. The dataset used was relatively small at only 1,000 observations. A higher number could increase accuracy and show new observations especially for terminal nodes with fewer observations. Second, the dataset was 22 years old and based off of an unknown German bank and so any future studies could benefit from a newer dataset based of the population of interest to that analysis. Third, more variables could be introduced such as annual income.

**References**

Han, Kamber, and Pei (2011). Data Mining: Concepts and Techniques, Third Edition Retrieved September 14, 2018 from http://hanj.cs.illinois.edu/cs412/bk3/01.pdf

Gonzalez-Garcia, J. (n.d.). Credit card delinquency statistics. Retrieved from https://www.creditcards.com/credit-card-news/credit-card-delinquency-statistics-1276.php

Staff, A. (2018, March 16). U.S. Foreclosure Activity Drops to 12-Year Low in 2017. Retrieved from https://www.attomdata.com/news/foreclosure-trends/2017-year-end-u-s-foreclosure-market-report/

Table 2 Distribution of Categorical Variables

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | | Personal Status | |
| Radio/TV | 280 | Female | 310 |
| new car | 234 | Male Divorced/Separated | 50 |
| furniture/equipment | 181 |
| used car | 103 | Male Married/widowed | 92 |
| business | 97 |
| education | 50 | Male Single | 548 |
| Other | 55 |
| Saving Status | | Employment | |
| <100 | 603 | <1 | 172 |
| >=1000 | 48 | >=7 | 253 |
| 100<=x<500 | 103 | 1<=X<4 | 339 |
| 500<=x<1000 | 63 | 4<=x<7 | 174 |
| no known savings | 183 | unemployed | 62 |
| Other Parties | | Property Magnitude | |
| Co applicant | 41 | Car | 332 |
| Guarantor | 52 | Life Insurance | 232 |
| None | 907 | No Property | 154 |
| Real Estate | 282 |
| Other Payment Plans | | Housing | |
| bank | 139 | For Free | 108 |
| none | 814 | Own | 713 |
| stores | 47 | Rent | 179 |
|
| Checking Status | | Credit History | |
| <0 | 274 | All paid | 49 |
| >=200 | 63 | Critical | 293 |
| 0<=x<200 | 269 | Previously Delayed | 88 |

Figure 1. Box plot of Credit Amount of Loan

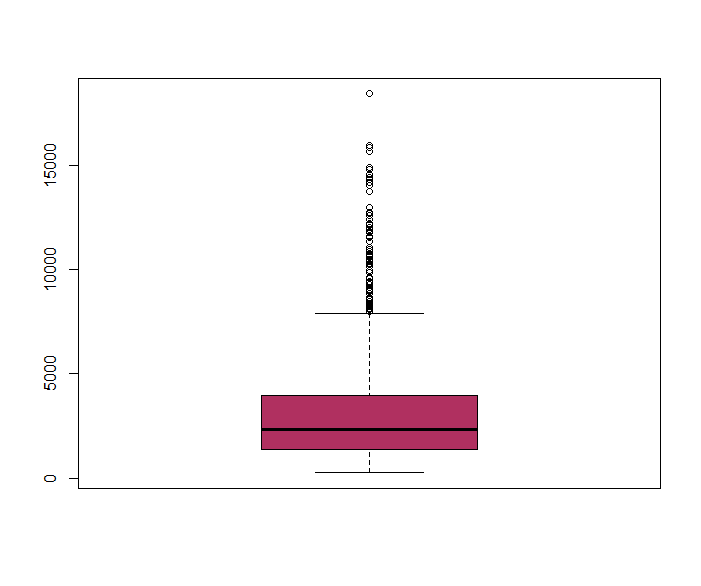


Figure 2. Boxplot of Age of Applicant and Duration of Loan

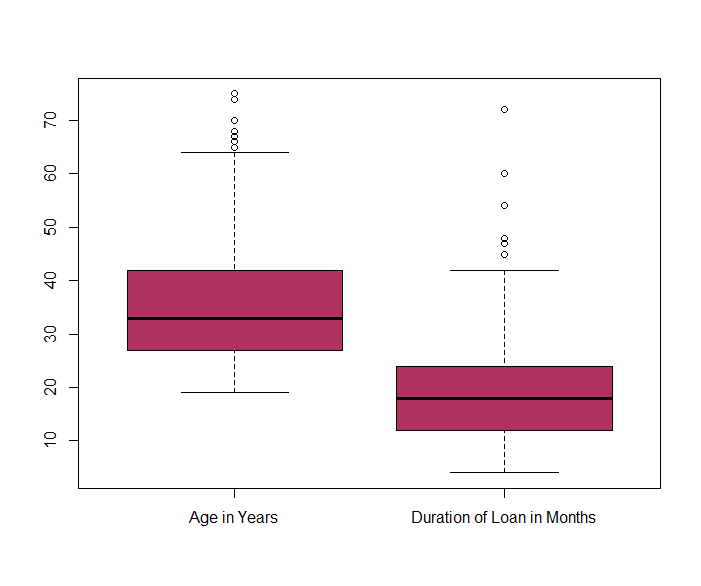


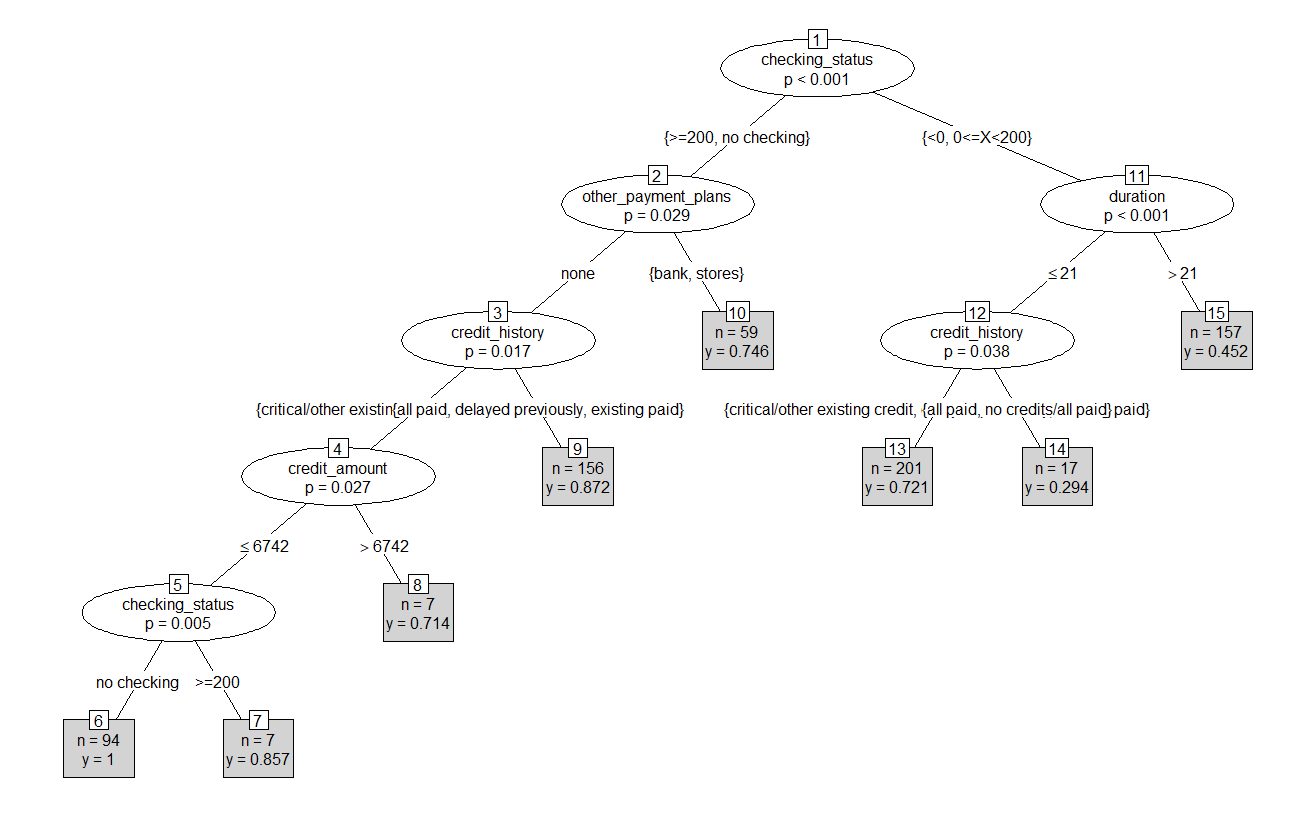
Figure 3. Visualization of Credit Decision Tree 

Figure 4. ROC Curve of Decision Tree

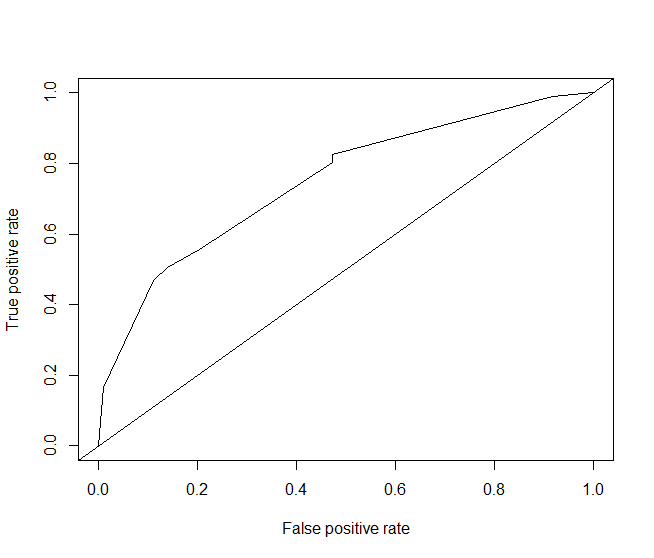


Table 4 - Confusion Matrix of Model 2

|  |  |  |
| --- | --- | --- |
| Model 2 - Threshold .75 Confusion Matrix | | |
|  | Predicted False | Predicted Positive |
| Actual False | 93 | 15 |
| Actual Positive | 96 | 98 |
| Error Rate | Accuracy | Sensitivity |
| 36.75% | 63.25% | 50.52% |
| Specificity | Precision | False Positive Rate |
| 48.69% | 86.73% | 13.89% |
| Score | | -171 |

Table 5- Confusion Matrix of Model 3

|  |  |  |
| --- | --- | --- |
| Model 3 - Threshold .35 Confusion Matrix | | |
|  | Predicted False | Predicted Positive |
| Actual False | 9 | 99 |
| Actual Positive | 2 | 192 |
| Error Rate | Accuracy | Sensitivity |
| 33.44% | 66.56% | 98.97% |
| Specificity | Precision | False Positive Rate |
| 4.48% | 65.98% | 91.67% |
| Score | | -497 |

**Appendix 1**

**Decision Tree Rules**

Conditional inference tree with 8 terminal nodes

Response: class

Inputs: checking\_status, duration, credit\_history, purpose, credit\_amount, savings\_status, employment, installment\_commitment, personal\_status, other\_parties, residence\_since, property\_magnitude, age, other\_payment\_plans, housing, existing\_credits, job, num\_dependents, own\_telephone, foreign\_worker

Number of observations: 698

1) checking\_status == {>=200, no checking}; criterion = 1, statistic = 79.224

2) other\_payment\_plans == {none}; criterion = 0.971, statistic = 17.699

3) credit\_history == {critical/other existing credit, no credits/all paid}; criterion = 0.983, statistic = 20.756

4) credit\_amount <= 6742; criterion = 0.973, statistic = 10.255

5) checking\_status == {no checking}; criterion = 0.995, statistic = 13.429

6)\* weights = 94

5) checking\_status == {>=200}

7)\* weights = 7

4) credit\_amount > 6742

8)\* weights = 7

3) credit\_history == {all paid, delayed previously, existing paid}

9)\* weights = 156

2) other\_payment\_plans == {bank, stores}

10)\* weights = 59

1) checking\_status == {<0, 0<=X<200}

11) duration <= 21; criterion = 1, statistic = 25.153

12) credit\_history == {critical/other existing credit, delayed previously, existing paid}; criterion = 0.962, statistic = 17.015

13)\* weights = 201

12) credit\_history == {all paid, no credits/all paid}

14)\* weights = 17

11) duration > 21

15)\* weights = 157

**Appendix 2**

**Code output of R.**

> # Assignment 3

> # by Kenneth Lulie, Data 630 - Ami Gates

> # Created 10/22/2018

> #Worked on 10/26/2018 through 10/28/2018

>

> ###Initial commands to set up session

>

>

> #Standard introductory screenshots

> Sys.time()

[1] "2018-10-28 17:03:08 EDT"

> Sys.info()

sysname release version nodename machine login user effective\_user

"Windows" ">= 8 x64" "build 9200" "LAPTOP-QHQ0RPOH" "x86-64" "Kenneth" "Kenneth" "Kenneth"

> R.version

\_

platform x86\_64-w64-mingw32

arch x86\_64

os mingw32

system x86\_64, mingw32

status

major 3

minor 5.1

year 2018

month 07

day 02

svn rev 74947

language R

version.string R version 3.5.1 (2018-07-02)

nickname Feather Spray

>

> # load libraries, load data

> library("party")

>

> #Set Working DIrecty

> setwd("D:/UMUC/630/Week 7/")

>

>

> #load in german credit file

> credit<-read.csv(file="credit-g.csv", head=TRUE, sep=",")

>

> #preview the structure

> str(credit)

'data.frame': 1000 obs. of 21 variables:

$ checking\_status : Factor w/ 4 levels "<0",">=200","0<=X<200",..: 1 3 4 1 1 4 4 3 4 3 ...

$ duration : int 6 48 12 42 24 36 24 36 12 30 ...

$ credit\_history : Factor w/ 5 levels "all paid","critical/other existing credit",..: 2 4 2 4 3 4 4 4 4 2 ...

$ purpose : Factor w/ 10 levels "business","domestic appliance",..: 7 7 3 4 5 3 4 10 7 5 ...

$ credit\_amount : int 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...

$ savings\_status : Factor w/ 5 levels "<100",">=1000",..: 5 1 1 1 1 5 4 1 2 1 ...

$ employment : Factor w/ 5 levels "<1",">=7","1<=X<4",..: 2 3 4 4 3 3 2 3 4 5 ...

$ installment\_commitment: int 4 2 2 2 3 2 3 2 2 4 ...

$ personal\_status : Factor w/ 4 levels "female div/dep/mar",..: 4 1 4 4 4 4 4 4 2 3 ...

$ other\_parties : Factor w/ 3 levels "co applicant",..: 3 3 3 2 3 3 3 3 3 3 ...

$ residence\_since : int 4 2 3 4 4 4 4 2 4 2 ...

$ property\_magnitude : Factor w/ 4 levels "car","life insurance",..: 4 4 4 2 3 3 2 1 4 1 ...

$ age : int 67 22 49 45 53 35 53 35 61 28 ...

$ other\_payment\_plans : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2 2 2 2 2 ...

$ housing : Factor w/ 3 levels "for free","own",..: 2 2 2 1 1 1 2 3 2 2 ...

$ existing\_credits : int 2 1 1 1 2 1 1 1 1 2 ...

$ job : Factor w/ 4 levels "high qualif/self emp/mgmt",..: 2 2 4 2 2 4 2 1 4 1 ...

$ num\_dependents : int 1 1 2 2 2 2 1 1 1 1 ...

$ own\_telephone : Factor w/ 2 levels "none","yes": 2 1 1 1 1 2 1 2 1 1 ...

$ foreign\_worker : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...

$ class : Factor w/ 2 levels "bad","good": 2 1 2 2 1 2 2 2 2 1 ...

>

> #Don't see any values or data types that need changing

>

> #Change bad to 0 and good to 1 for ROC curve later

> #Converts to char from factor, then sets bad to 0 and good to 1, then sets it to num from char

> credit$class <- as.character(credit$class)

> credit$class[credit$class == "bad"] <- 0

> credit$class[credit$class == "good"] <- 1

> credit$class <- as.numeric(credit$class)

>

>

>

>

>

> #Check data quality, dont see any NAs in the data

> apply(credit, 2, function (credit) sum(is.na(credit)))

checking\_status duration credit\_history purpose credit\_amount savings\_status

0 0 0 0 0 0

employment installment\_commitment personal\_status other\_parties residence\_since property\_magnitude

0 0 0 0 0 0

age other\_payment\_plans housing existing\_credits job num\_dependents

0 0 0 0 0 0

own\_telephone foreign\_worker class

0 0 0

>

> #Review data with summary

> #No obvious problems, good 1000 points in dataset which is good.

> summary(credit)

checking\_status duration credit\_history purpose credit\_amount savings\_status

<0 :274 Min. : 4.0 all paid : 49 radio/tv :280 Min. : 250 <100 :603

>=200 : 63 1st Qu.:12.0 critical/other existing credit:293 new car :234 1st Qu.: 1366 >=1000 : 48

0<=X<200 :269 Median :18.0 delayed previously : 88 furniture/equipment:181 Median : 2320 100<=X<500 :103

no checking:394 Mean :20.9 existing paid :530 used car :103 Mean : 3271 500<=X<1000 : 63

3rd Qu.:24.0 no credits/all paid : 40 business : 97 3rd Qu.: 3972 no known savings:183

Max. :72.0 education : 50 Max. :18424

(Other) : 55

employment installment\_commitment personal\_status other\_parties residence\_since property\_magnitude age

<1 :172 Min. :1.000 female div/dep/mar:310 co applicant: 41 Min. :1.000 car :332 Min. :19.00

>=7 :253 1st Qu.:2.000 male div/sep : 50 guarantor : 52 1st Qu.:2.000 life insurance :232 1st Qu.:27.00

1<=X<4 :339 Median :3.000 male mar/wid : 92 none :907 Median :3.000 no known property:154 Median :33.00

4<=X<7 :174 Mean :2.973 male single :548 Mean :2.845 real estate :282 Mean :35.55

unemployed: 62 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:42.00

Max. :4.000 Max. :4.000 Max. :75.00

other\_payment\_plans housing existing\_credits job num\_dependents own\_telephone foreign\_worker class

bank :139 for free:108 Min. :1.000 high qualif/self emp/mgmt:148 Min. :1.000 none:596 no : 37 Min. :0.0

none :814 own :713 1st Qu.:1.000 skilled :630 1st Qu.:1.000 yes :404 yes:963 1st Qu.:0.0

stores: 47 rent :179 Median :1.000 unemp/unskilled non res : 22 Median :1.000 Median :1.0

Mean :1.407 unskilled resident :200 Mean :1.155 Mean :0.7

3rd Qu.:2.000 3rd Qu.:1.000 3rd Qu.:1.0

Max. :4.000 Max. :2.000 Max. :1.0

>

>

>

> ###Make boxplots

> boxplot(credit$credit\_amount, col="maroon")

>

> boxplot(credit$age, credit$duration, col="maroon",names=c("Age in Years","Duration of Loan in Months"))

>

>

>

>

> ### reused commands from exercise, standard 70 30% split

> #split the data into a training and test set

> set.seed(1234)

> ind <- sample(2, nrow(credit), replace = TRUE, prob = c(0.7, 0.3))

> train.data <- credit[ind == 1, ]

> test.data <- credit[ind == 2, ]

>

>

>

> ## reused commands from excercise, predit on class and use all variables.

> #6. Run the method on a training data

> myFormula<-class~.

> credit\_ctree <- ctree(myFormula, data = train.data)

>

> ##Reuse commands to print the tree structure

> #7. output the tree structure

> print(credit\_ctree)

Conditional inference tree with 8 terminal nodes

Response: class

Inputs: checking\_status, duration, credit\_history, purpose, credit\_amount, savings\_status, employment, installment\_commitment, personal\_status, other\_parties, residence\_since, property\_magnitude, age, other\_payment\_plans, housing, existing\_credits, job, num\_dependents, own\_telephone, foreign\_worker

Number of observations: 698

1) checking\_status == {>=200, no checking}; criterion = 1, statistic = 79.224

2) other\_payment\_plans == {none}; criterion = 0.971, statistic = 17.699

3) credit\_history == {critical/other existing credit, no credits/all paid}; criterion = 0.983, statistic = 20.756

4) credit\_amount <= 6742; criterion = 0.973, statistic = 10.255

5) checking\_status == {no checking}; criterion = 0.995, statistic = 13.429

6)\* weights = 94

5) checking\_status == {>=200}

7)\* weights = 7

4) credit\_amount > 6742

8)\* weights = 7

3) credit\_history == {all paid, delayed previously, existing paid}

9)\* weights = 156

2) other\_payment\_plans == {bank, stores}

10)\* weights = 59

1) checking\_status == {<0, 0<=X<200}

11) duration <= 21; criterion = 1, statistic = 25.153

12) credit\_history == {critical/other existing credit, delayed previously, existing paid}; criterion = 0.962, statistic = 17.015

13)\* weights = 201

12) credit\_history == {all paid, no credits/all paid}

14)\* weights = 17

11) duration > 21

15)\* weights = 157

>

>

> #8. visualize the tree

> nodes(credit\_ctree, 2)

[[1]]

2) other\_payment\_plans == {none}; criterion = 0.971, statistic = 17.699

3) credit\_history == {critical/other existing credit, no credits/all paid}; criterion = 0.983, statistic = 20.756

4) credit\_amount <= 6742; criterion = 0.973, statistic = 10.255

5) checking\_status == {no checking}; criterion = 0.995, statistic = 13.429

6)\* weights = 94

5) checking\_status == {>=200}

7)\* weights = 7

4) credit\_amount > 6742

8)\* weights = 7

3) credit\_history == {all paid, delayed previously, existing paid}

9)\* weights = 156

2) other\_payment\_plans == {bank, stores}

10)\* weights = 59

> plot(credit\_ctree)

> plot(credit\_ctree, type="simple")

>

>

> #10. Evaluate the model on a test data

> testPred <- predict(credit\_ctree, newdata = test.data)

>

>

>

> ### Get AUC number

> library(pROC)

> auc(test.data$class, testPred)

Area under the curve: 0.7498

Warning message:

In roc.default(response, predictor, auc = TRUE, ...) :

Deprecated use a matrix as predictor. Unexpected results may be produced, please pass a numeric vector.

> #.7498

>

>

>

> ### plot AUC Chart

> library(ROCR)

>

> # explanation of order prediction(predictions, labels, label.ordering = NULL)

> #so you want to create a object with the prediction data called pred.

> #The first agrument hte prediction takes, is the continous numbers generated for the predictions by the ctree

> #The second agrument is the label or 'the truth'.

> pred <- prediction(testPred, test.data$class, label.ordering = NULL)

>

> #Plot it out, measure tpr, xmeasure fpr

> roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")

> plot(roc.perf)

> abline(a=0, b= 1)

>

> ## I believe this is indicating that we can get 50% of the true positives by using a 20% false positive cutoff.

> ## we can get about 75% of the true positive, by allowing a 40% false positive cutoff.

>

>

>

>

> ## Confuson Matrix with a normal 50% split.

> table (round(testPred), test.data$class)

0 1

0 57 34

1 51 160

> # get spec to sens ratio

> # 0 1

> #0 57 34

> #1 51 160

> #-51\*5 + -34 = -289

>

>

> ## This is a only 75% and up cutoff

> strictcutoff <- testPred

> strictcutoff[strictcutoff<.75] <- 0

> strictcutoff[strictcutoff >.75] <- 1

> table (strictcutoff, test.data$class)

strictcutoff 0 1

0 93 96

1 15 98

> ##I think top is actual, and side is predictions

> # so for this there would be 15 predicted as good, but actually bad, and 96 predicted bad but actually good

> # 0 1

> #0 93 96

> #1 15 98

>

> #-15\*5 + -96

> #-171

>

>

> ##This model uses a 35% and up cutoff

> relaxcutoff <- testPred

> relaxcutoff[relaxcutoff <.35] <- 0

> relaxcutoff[relaxcutoff >.35] <- 1

>

> table (relaxcutoff, test.data$class)

relaxcutoff 0 1

0 9 2

1 99 192

> # 0 1

> #0 9 2

> #1 99 192

> # -99 \* 5 + -2 = -497