Keosotra Veng
Stat 6620

Homework 4 Part 1

Tree Based Algorithm

Step 1: Step 1 - collecting data

- We got the data from:http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Step 2: Exploring and preparing the data ----

-Import the credit.csv data set to R

```
credit <- read.csv("credit.csv")
str(credit)
```

```
## 'data.frame':    1000 obs. of  17 variables:
##  $ checking_balance    : Factor w/ 4 levels "< 0 DM","> 200 DM",..: 1 3 4
1 1 4 4 3 4 3 ...
##  $ months_loan_duration: int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history      : Factor w/ 5 levels "critical","good",..: 1 2 1 2
4 2 2 2 2 1 ...
##  $ purpose             : Factor w/ 6 levels "business","car",..: 5 5 4 5 2
4 5 2 5 2 ...
##  $ amount              : int  1169 5951 2096 7882 4870 9055 2835 6948 3059
5234 ...
##  $ savings_balance     : Factor w/ 5 levels "< 100 DM","> 1000 DM",..: 5 1
1 1 1 5 4 1 2 1 ...
##  $ employment_duration : Factor w/ 5 levels "< 1 year","> 7 years",..: 2 3
4 4 3 3 2 3 4 5 ...
##  $ percent_of_income   : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ years_at_residence  : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ age                 : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ other_credit        : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2
2 2 2 2 ...
##  $ housing             : Factor w/ 3 levels "other","own",..: 2 2 2 1 1 1
2 3 2 2 ...
##  $ existing_loans_count: int  2 1 1 1 2 1 1 1 1 2 ...
##  $ job                 : Factor w/ 4 levels "management","skilled",..: 2 2
4 2 2 4 2 1 4 1 ...
##  $ dependents          : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ phone               : Factor w/ 2 levels "no","yes": 2 1 1 1 1 2 1 2 1
1 ...
##  $ default             : Factor w/ 2 levels "no","yes": 1 2 1 1 2 1 1 1 1
2 ...
```

- We look at checking_balance and savings_balance columns.

Keosotra Veng
Stat 6620

```
table(credit$checking_balance)
```

```
##
##     < 0 DM   > 200 DM 1 - 200 DM    unknown
##        274          63         269         394
```

```
table(credit$savings_balance)
```

```
##
##      < 100 DM     > 1000 DM  100 - 500 DM 500 - 1000 DM       unknown
##           603            48           103            63           183
```

- Looking at the summarry statistics of months_loan_duration and amount of loan.

```
summary(credit$months_loan_duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0    12.0    18.0    20.9    24.0    72.0
```

```
summary(credit$amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     250    1366    2320    3271    3972   18420
```

- We look how many default and not default in the loan. 700 not default and 300 default.

```
table(credit$default)
```

```
##
##  no yes
## 700 300
```

- We randomize our sample.

```
set.seed(123)
train_sample <- sample(1000, 900)
str(train_sample)
```

```
##  int [1:900] 288 788 409 881 937 46 525 887 548 453 ...
```

- We split the data 900 into training and 100 into testing.

```
credit_train <- credit[train_sample, ]
credit_test  <- credit[-train_sample, ]
```

- we look at the default numbers in our train and test data.

```
prop.table(table(credit_train$default))
```

```
##
##        no       yes
## 0.7033333 0.2966667
```

```
prop.table(table(credit_test$default))
```

```
##
##   no  yes
## 0.67 0.33
```

Step 3: Training a model on the data ----

- We use C50 to build the simplest decision tree

```
library(C50)
credit_model <- C5.0(credit_train[-17], credit_train$default)
```

- Display simple facts about the tree. The tree size is 57.

```
credit_model
```

```
##
## Call:
## C5.0.default(x = credit_train[-17], y = credit_train$default)
##
## Classification Tree
## Number of samples: 900
## Number of predictors: 16
##
## Tree size: 57
##
## Non-standard options: attempt to group attributes
```

- Display detailed information about the tree. The model only correctly predicted 767 out of 900 observations.

```
summary(credit_model)
```

```
##
## Call:
## C5.0.default(x = credit_train[-17], y = credit_train$default)
##
##
## C5.0 [Release 2.07 GPL Edition]      Wed May 03 02:06:03 2017
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 900 cases (17 attributes) from undefined.data
##
## Decision tree:
##
## checking_balance in {> 200 DM,unknown}: no (412/50)
## checking_balance in {< 0 DM,1 - 200 DM}:
## :...credit_history in {perfect,very good}: yes (59/18)
##     credit_history in {critical,good,poor}:
##     :...months_loan_duration <= 22:
##         :...credit_history = critical: no (72/14)
##         :   credit_history = poor:
```

```
##         :   :...dependents > 1: no (5)
##         :   :   dependents <= 1:
##         :   :   :...years_at_residence <= 3: yes (4/1)
##         :   :       years_at_residence > 3: no (5/1)
##         :   credit_history = good:
##         :   :...savings_balance in {> 1000 DM,500 - 1000 DM}: no (15/1)
##         :       savings_balance = 100 - 500 DM:
##         :       :...other_credit = bank: yes (3)
##         :       :   other_credit in {none,store}: no (9/2)
##         :       savings_balance = unknown:
##         :       :...other_credit = bank: yes (1)
##         :       :   other_credit in {none,store}: no (21/8)
##         :       savings_balance = < 100 DM:
##         :       :...purpose in {business,car0,renovations}: no (8/2)
##         :           purpose = education:
##         :           :...checking_balance = < 0 DM: yes (4)
##         :           :   checking_balance = 1 - 200 DM: no (1)
##         :           purpose = car:
##         :           :...employment_duration = > 7 years: yes (5)
##         :           :   employment_duration = unemployed: no (4/1)
##         :           :   employment_duration = < 1 year:
##         :           :   :...years_at_residence <= 2: yes (5)
##         :           :   :   years_at_residence > 2: no (3/1)
##         :           :   employment_duration = 1 - 4 years:
##         :           :   :...years_at_residence <= 2: yes (2)
##         :           :   :   years_at_residence > 2: no (6/1)
##         :           :   employment_duration = 4 - 7 years:
##         :           :   :...amount <= 1680: yes (2)
##         :           :       amount > 1680: no (3)
##         :           purpose = furniture/appliances:
##         :           :...job in {management,unskilled}: no (23/3)
##         :               job = unemployed: yes (1)
##         :               job = skilled:
##         :               :...months_loan_duration > 13: [S1]
##         :                   months_loan_duration <= 13:
##         :                   :...housing in {other,own}: no (23/4)
##         :                       housing = rent:
##         :                       :...percent_of_income <= 3: yes (3)
##         :                           percent_of_income > 3: no (2)
##         months_loan_duration > 22:
##         :...savings_balance = > 1000 DM: no (2)
##             savings_balance = 500 - 1000 DM: yes (4/1)
##             savings_balance = 100 - 500 DM:
##             :...credit_history in {critical,poor}: no (14/3)
##             :   credit_history = good:
##             :   :...other_credit = bank: no (1)
##             :       other_credit in {none,store}: yes (12/2)
##             savings_balance = unknown:
##             :...checking_balance = 1 - 200 DM: no (17)
```

```
##             :   checking_balance = < 0 DM:
##             :   :...credit_history = critical: no (1)
##             :       credit_history in {good,poor}: yes (12/3)
##             savings_balance = < 100 DM:
##             :...months_loan_duration > 47: yes (21/2)
##                 months_loan_duration <= 47:
##                 :...housing = other:
##                     :...percent_of_income <= 2: no (6)
##                     :   percent_of_income > 2: yes (9/3)
##                     housing = rent:
##                     :...other_credit = bank: no (1)
##                     :   other_credit in {none,store}: yes (16/3)
##                     housing = own:
##                     :...employment_duration = > 7 years: no (13/4)
##                         employment_duration = 4 - 7 years:
##                         :...job in {management,skilled,
##                         :   :       unemployed}: yes (9/1)
##                         :   job = unskilled: no (1)
##                         employment_duration = unemployed:
##                         :...years_at_residence <= 2: yes (4)
##                         :   years_at_residence > 2: no (3)
##                         employment_duration = 1 - 4 years:
##                         :...purpose in {business,car0,education}: yes (7/1
)
##                         :   purpose in {furniture/appliances,
##                         :   :           renovations}: no (7)
##                         :   purpose = car:
##                         :   :...years_at_residence <= 3: yes (3)
##                         :       years_at_residence > 3: no (3)
##                         employment_duration = < 1 year:
##                         :...years_at_residence > 3: yes (5)
##                             years_at_residence <= 3:
##                             :...other_credit = bank: no (0)
##                                 other_credit = store: yes (1)
##                                 other_credit = none:
##                                 :...checking_balance = 1 - 200 DM: no (8/2
)
##                                     checking_balance = < 0 DM:
##                                     :...job in {management,skilled,
##                                         :       unemployed}: yes (2)
##                                         job = unskilled: no (3/1)
##
## SubTree [S1]
##
## employment_duration in {< 1 year,4 - 7 years}: no (4)
## employment_duration in {> 7 years,1 - 4 years,unemployed}: yes (10)
##
##
## Evaluation on training data (900 cases):
```

```
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      56  133(14.8%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     598    35    (a): class no
##      98   169    (b): class yes
##
##
##  Attribute usage:
##
##  100.00% checking_balance
##   54.22% credit_history
##   47.67% months_loan_duration
##   38.11% savings_balance
##   14.33% purpose
##   14.33% housing
##   12.56% employment_duration
##    9.00% job
##    8.67% other_credit
##    6.33% years_at_residence
##    2.22% percent_of_income
##    1.56% dependents
##    0.56% amount
##
##
## Time: 0.0 secs
```

Step 4: Evaluating model performance ----

- Create a factor vector of predictions on test data

```
credit_pred <- predict(credit_model, credit_test)
```

- Cross tabulation of predicted versus actual classes. We use our model on the test data and we only correctly predicted 73 out of 100 observations.
- We predicted 19 of the observations to be not default loans but it's actualy default loans and we predicted 8 to be default loans but they are not default loans.

```
library(gmodels)
CrossTable(credit_test$default, credit_pred,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('actual default', 'predicted default'))

##
##
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                 | predicted default
## actual default |        no |       yes | Row Total |
## ---------------|-----------|-----------|-----------|
##             no |        59 |         8 |        67 |
##                |     0.590 |     0.080 |           |
## ---------------|-----------|-----------|-----------|
##            yes |        19 |        14 |        33 |
##                |     0.190 |     0.140 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |        78 |        22 |       100 |
## ---------------|-----------|-----------|-----------|
##
##
```

Step 5: Improving model performance ----

- We use boosting to improve the accuracy of decision trees
- Boosted decision tree with 10 trials
- Our result is slightly better. Now we correctly predicted 82 out of 100 observations.

```
credit_boost10 <- C5.0(credit_train[-17], credit_train$default,
                       trials = 10)
credit_boost10
```

```
##
## Call:
## C5.0.default(x = credit_train[-17], y = credit_train$default, trials = 10)
##
## Classification Tree
## Number of samples: 900
## Number of predictors: 16
##
## Number of boosting iterations: 10
## Average tree size: 47.5
##
## Non-standard options: attempt to group attributes
```

```
credit_boost_pred10 <- predict(credit_boost10, credit_test)
CrossTable(credit_test$default, credit_boost_pred10,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('actual default', 'predicted default'))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                | predicted default
## actual default |        no |       yes | Row Total |
## ---------------|-----------|-----------|-----------|
##             no |        62 |         5 |        67 |
##                |     0.620 |     0.050 |           |
## ---------------|-----------|-----------|-----------|
##            yes |        13 |        20 |        33 |
##                |     0.130 |     0.200 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |        75 |        25 |       100 |
## ---------------|-----------|-----------|-----------|
##
##
```

Predicting not default while it turns out default is very costly, so we want to build in some cost into the model.

- Create dimensions for a cost matrix

```
matrix_dimensions <- list(c("no", "yes"), c("no", "yes"))
names(matrix_dimensions) <- c("predicted", "actual")
matrix_dimensions
```

```
## $predicted
## [1] "no"  "yes"
##
## $actual
## [1] "no"  "yes"
```

- Build the matrix

```
error_cost <- matrix(c(0, 1, 4, 0), nrow = 2, dimnames = matrix_dimensions)
error_cost
```

```
##          actual
## predicted no yes
##       no   0   4
##       yes  1   0
```

- Apply the cost matrix to the tree

- With the cost matrix our result is worse, we only correctly predicted 63 out of 100 observations but we greatly reduce the incorrect prediction of predicting not default but the actual loan is default down to 7 but it costs us to have lower prediction in return.

```
credit_cost <- C5.0(credit_train[-17], credit_train$default,
                          costs = error_cost)
credit_cost_pred <- predict(credit_cost, credit_test)

CrossTable(credit_test$default, credit_cost_pred,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('actual default', 'predicted default'))

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                | predicted default
## actual default |        no |       yes | Row Total |
## ---------------|-----------|-----------|-----------|
##             no |        37 |        30 |        67 |
##                |     0.370 |     0.300 |           |
## ---------------|-----------|-----------|-----------|
##            yes |         7 |        26 |        33 |
##                |     0.070 |     0.260 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |        44 |        56 |       100 |
## ---------------|-----------|-----------|-----------|
##
##
```