PeerReview_5

Minxin Cheng

```r
#install.packages("C50")
#install.packages("RWeka")
library(C50)
library(gmodels)
library(RWeka)
```

Problem 1

1. Read in data file

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

2. Get the overall information of the dataset

```r
# check the structure of the dataset
str(credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ checking_balance    : chr  "< 0 DM" "1 - 200 DM" "unknown" "< 0 DM" ...
##  $ months_loan_duration: int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history      : chr  "critical" "repaid" "critical" "repaid" ...
##  $ purpose             : chr  "radio/tv" "radio/tv" "education" "furniture" ...
##  $ amount              : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ savings_balance     : chr  "unknown" "< 100 DM" "< 100 DM" "< 100 DM" ...
##  $ employment_length   : chr  "> 7 yrs" "1 - 4 yrs" "4 - 7 yrs" "4 - 7 yrs" ...
##  $ installment_rate    : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status     : chr  "single male" "female" "single male" "single male" ...
##  $ other_debtors       : chr  "none" "none" "none" "guarantor" ...
##  $ residence_history   : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property            : chr  "real estate" "real estate" "real estate" "building society savings" .
##  $ age                 : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ installment_plan    : chr  "none" "none" "none" "none" ...
##  $ housing             : chr  "own" "own" "own" "for free" ...
##  $ existing_credits    : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ default             : int  1 2 1 1 2 1 1 1 1 2 ...
##  $ dependents          : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone           : chr  "yes" "none" "none" "none" ...
##  $ foreign_worker      : chr  "yes" "yes" "yes" "yes" ...
##  $ job                 : chr  "skilled employee" "skilled employee" "unskilled resident" "skilled em
```

```r
# summary features see if there is anything stands out
table(credit$checking_balance)
```

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

```r
table(credit$savins_balance)
```

```
## < table of extent 0 >
```

```r
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
```

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     250    1366    2320    3271    3972   18424
```

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

```
## 
##   1   2 
## 700 300
```

3. Split data

```r
# randomly generate 900 numbers
set.seed(123)
train_sample <- sample(1000, 900)
str(train_sample)
```

```
##  int [1:900] 415 463 179 526 195 938 818 118 299 229 ...
```

```r
# extract rows of sample number as train data, remaining data as test data
credit_train <- credit[train_sample, ]
credit_test <- credit[-train_sample, ]
# check the propotion of the default column in both training and testing data
prop.table(table(credit_train$default))
```

```
## 
##         1         2 
## 0.7055556 0.2944444
```

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

```
##
##    1    2
## 0.65 0.35
```

4. Make prediction

```r
# convert default column as factor
credit_train$default <- as.factor(credit_train$default)
# build the classifer
credit_model <- C5.0(credit_train[-17],
                     credit_train$default)
credit_model
```

```
##
## Call:
## C5.0.default(x = credit_train[-17], y = credit_train$default)
##
## Classification Tree
## Number of samples: 900
## Number of predictors: 20
##
## Tree size: 42
##
## Non-standard options: attempt to group attributes
```

```r
summary(credit_model)
```

```
##
## Call:
## C5.0.default(x = credit_train[-17], y = credit_train$default)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Oct 25 17:04:41 2020
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 900 cases (21 attributes) from undefined.data
##
## Decision tree:
##
## checking_balance in {unknown,> 200 DM}: 1 (412/54)
## checking_balance in {< 0 DM,1 - 200 DM}:
## :...credit_history in {fully repaid this bank,fully repaid}:
##     :...housing = rent: 2 (16/1)
##     :   housing = for free:
##     :   :...other_debtors in {none,guarantor}: 2 (12/1)
##     :   :   other_debtors = co-applicant: 1 (2)
##     :   housing = own:
```

```
##     :   :...purpose in {radio/tv,education,repairs,domestic appliances,
##     :       :           others}: 2 (6/1)
##     :       purpose in {car (used),business,retraining}: 1 (10/2)
##     :       purpose = car (new):
##     :       :...months_loan_duration <= 22: 2 (6)
##     :       :   months_loan_duration > 22: 1 (2)
##     :       purpose = furniture:
##     :       :...installment_plan in {none,stores}: 1 (4)
##     :           installment_plan = bank: 2 (5/1)
##     credit_history in {repaid,critical,delayed}:
##     :...months_loan_duration <= 15: 1 (180/45)
##         months_loan_duration > 15:
##         :...savings_balance in {unknown,> 1000 DM}:
##             :...credit_history in {critical,delayed}: 1 (14)
##             :   credit_history = repaid:
##             :   :...purpose = car (new): 2 (7/1)
##             :       purpose in {business,education,repairs,domestic appliances,
##             :       :           retraining,others}: 1 (5)
##             :       purpose = furniture:
##             :       :...age <= 27: 2 (2)
##             :       :   age > 27: 1 (5)
##             :       purpose = radio/tv:
##             :       :...amount <= 6110: 1 (5)
##             :       :   amount > 6110: 2 (2)
##             :       purpose = car (used):
##             :       :...amount <= 6967: 1 (4)
##             :           amount > 6967: 2 (2)
##             savings_balance in {< 100 DM,101 - 500 DM,501 - 1000 DM}:
##             :...months_loan_duration > 47: 2 (23/3)
##                 months_loan_duration <= 47:
##                 :...employment_length = 0 - 1 yrs:
##                     :...residence_history <= 1: 1 (16/6)
##                     :   residence_history > 1: 2 (27/6)
##                     employment_length = unemployed:
##                     :...residence_history <= 2: 2 (7)
##                     :   residence_history > 2: 1 (12/2)
##                     employment_length = > 7 yrs:
##                     :...purpose = car (new): 2 (11/3)
##                     :   purpose in {radio/tv,car (used),education,repairs,
##                     :   :           domestic appliances,retraining,
##                     :   :           others}: 1 (13/1)
##                     :   purpose = furniture:
##                     :   :...job in {skilled employee,unskilled resident,
##                     :   :   :       unemployed non-resident}: 1 (5/1)
##                     :   :   job = mangement self-employed: 2 (2)
##                     :   purpose = business:
##                     :   :...personal_status in {female,divorced male}: 2 (3)
##                     :       personal_status in {single male,
##                     :                           married male}: 1 (3)
##                     employment_length = 1 - 4 yrs:
##                     :...installment_rate > 3: 2 (20/3)
##                     :   installment_rate <= 3:
##                     :   :...other_debtors = guarantor: 1 (2)
##                     :       other_debtors = co-applicant: 2 (3)
##                     :       other_debtors = none:
##                     :       :...checking_balance = 1 - 200 DM: 1 (8/1)
##                     :           checking_balance = < 0 DM: [S1]
##                     employment_length = 4 - 7 yrs:
##                     :...savings_balance in {101 - 500 DM,
##                         :                   501 - 1000 DM}: 1 (8)
##                         savings_balance = < 100 DM:
##                         :...job in {mangement self-employed,unskilled resident,
##                             :       unemployed non-resident}: 1 (6)
##                             job = skilled employee:
##                             :...dependents > 1: 1 (3/1)
##                                 dependents <= 1:
##                                 :...months_loan_duration <= 22: 1 (3)
##                                     months_loan_duration > 22: 2 (8)
##
## SubTree [S1]
##
## personal_status in {female,single male}: 2 (13/3)
## personal_status in {married male,divorced male}: 1 (3)
##
##
## Evaluation on training data (900 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      42  136(15.1%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     612    23    (a): class 1
##     113   152    (b): class 2
##
##
##  Attribute usage:
##
##  100.00% checking_balance
##   54.22% credit_history
##   48.11% months_loan_duration
##   27.22% savings_balance
##   19.56% employment_length
##   11.33% purpose
##    7.00% housing
##    6.89% residence_history
##    5.44% installment_rate
##    4.78% other_debtors
##    3.00% job
##    2.44% personal_status
##    1.56% dependents
##    1.44% amount
##    1.00% installment_plan
##    0.78% age
```

```
##
##
## Time: 0.0 secs
```

```
# make prediction
credit_pred <- predict(credit_model,
                       credit_test)
```

5. Evaluate the prediction

```
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 |         1 |         2 | Row Total |
## ---------------|-----------|-----------|-----------|
##              1 |        55 |        10 |        65 |
##                |     0.550 |     0.100 |           |
## ---------------|-----------|-----------|-----------|
##              2 |        22 |        13 |        35 |
##                |     0.220 |     0.130 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |        77 |        23 |       100 |
## ---------------|-----------|-----------|-----------|
##
##
```

As the table shows, 100 prediction were made. The accuracy is (55 + 13) / 100 = 0.68.

6. Improve the model performance

```
# boost the accuracy of the tree
# the trials indicating the number of separate decision trees to use in the
# boosted team.
credit_boost10 <- C5.0(credit_train[-17],
                       credit_train$default, trials = 10)
summary(credit_boost10)
```

```
##
## Call:
## C5.0.default(x = credit_train[-17], y = credit_train$default, trials = 10)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Oct 25 17:04:41 2020
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 900 cases (21 attributes) from undefined.data
##
## -----  Trial 0:  -----
##
## Decision tree:
##
## checking_balance in {unknown,> 200 DM}: 1 (412/54)
## checking_balance in {< 0 DM,1 - 200 DM}:
## :...credit_history in {fully repaid this bank,fully repaid}:
##     :...housing = rent: 2 (16/1)
##     :   housing = for free:
##     :   :...other_debtors in {none,guarantor}: 2 (12/1)
##     :   :   other_debtors = co-applicant: 1 (2)
##     :   housing = own:
##     :   :...purpose in {radio/tv,education,repairs,domestic appliances,
##     :       :           others}: 2 (6/1)
##     :       purpose in {car (used),business,retraining}: 1 (10/2)
##     :       purpose = car (new):
##     :       :...months_loan_duration <= 22: 2 (6)
##     :       :   months_loan_duration > 22: 1 (2)
##     :       purpose = furniture:
##     :       :...installment_plan in {none,stores}: 1 (4)
##     :           installment_plan = bank: 2 (5/1)
##     credit_history in {repaid,critical,delayed}:
##     :...months_loan_duration <= 15: 1 (180/45)
##         months_loan_duration > 15:
##         :...savings_balance in {unknown,> 1000 DM}:
##             :...credit_history in {critical,delayed}: 1 (14)
##             :   credit_history = repaid:
##             :   :...purpose = car (new): 2 (7/1)
##             :       purpose in {business,education,repairs,domestic appliances,
##             :       :           retraining,others}: 1 (5)
##             :       purpose = furniture:
##             :       :...age <= 27: 2 (2)
##             :       :   age > 27: 1 (5)
##             :       purpose = radio/tv:
##             :       :...amount <= 6110: 1 (5)
##             :       :   amount > 6110: 2 (2)
##             :       purpose = car (used):
##             :       :...amount <= 6967: 1 (4)
##             :           amount > 6967: 2 (2)
##             savings_balance in {< 100 DM,101 - 500 DM,501 - 1000 DM}:
##             :...months_loan_duration > 47: 2 (23/3)
##                 months_loan_duration <= 47:
```

```
##                 :...employment_length = 0 - 1 yrs:
##                     :...residence_history <= 1: 1 (16/6)
##                     :   residence_history > 1: 2 (27/6)
##                     employment_length = unemployed:
##                     :...residence_history <= 2: 2 (7)
##                     :   residence_history > 2: 1 (12/2)
##                     employment_length = > 7 yrs:
##                     :...purpose = car (new): 2 (11/3)
##                     :   purpose in {radio/tv,car (used),education,repairs,
##                     :   :           domestic appliances,retraining,
##                     :   :           others}: 1 (13/1)
##                     :   purpose = furniture:
##                     :   :...job in {skilled employee,unskilled resident,
##                     :   :   :       unemployed non-resident}: 1 (5/1)
##                     :   :   job = mangement self-employed: 2 (2)
##                     :   purpose = business:
##                     :   :...personal_status in {female,divorced male}: 2 (3)
##                     :       personal_status in {single male,
##                     :                           married male}: 1 (3)
##                     employment_length = 1 - 4 yrs:
##                     :...installment_rate > 3: 2 (20/3)
##                     :   installment_rate <= 3:
##                     :   :...other_debtors = guarantor: 1 (2)
##                     :       other_debtors = co-applicant: 2 (3)
##                     :       other_debtors = none:
##                     :       :...checking_balance = 1 - 200 DM: 1 (8/1)
##                     :           checking_balance = < 0 DM: [S1]
##                     employment_length = 4 - 7 yrs:
##                     :...savings_balance in {101 - 500 DM,
##                         :                   501 - 1000 DM}: 1 (8)
##                         savings_balance = < 100 DM:
##                         :...job in {mangement self-employed,unskilled resident,
##                             :       unemployed non-resident}: 1 (6)
##                             job = skilled employee:
##                             :...dependents > 1: 1 (3/1)
##                                 dependents <= 1:
##                                 :...months_loan_duration <= 22: 1 (3)
##                                     months_loan_duration > 22: 2 (8)
##
## SubTree [S1]
##
## personal_status in {female,single male}: 2 (13/3)
## personal_status in {married male,divorced male}: 1 (3)
##
## -----  Trial 1:  -----
##
## Decision tree:
##
## checking_balance = unknown:
## :...installment_plan = none: 1 (264.1/49.6)
## :   installment_plan in {bank,stores}:
## :   :...other_debtors in {guarantor,co-applicant}: 1 (3.2)
## :       other_debtors = none:
## :       :...employment_length in {1 - 4 yrs,0 - 1 yrs,unemployed}: 2 (40.5/10.3)
## :           employment_length in {4 - 7 yrs,> 7 yrs}: 1 (29.3/8.6)
## checking_balance in {< 0 DM,1 - 200 DM,> 200 DM}:
## :...other_debtors = guarantor: 1 (35.3/7.2)
##     other_debtors in {none,co-applicant}:
##     :...savings_balance in {unknown,501 - 1000 DM,> 1000 DM}:
##         :...amount > 1530: 1 (63.4/13.8)
##         :   amount <= 1530:
##         :   :...installment_rate <= 2: 1 (6.4)
##         :       installment_rate > 2:
##         :       :...dependents > 1: 2 (5.1)
##         :           dependents <= 1:
##         :           :...months_loan_duration <= 11: 1 (5.6)
##         :               months_loan_duration > 11: 2 (25.6/7.9)
##         savings_balance in {< 100 DM,101 - 500 DM}:
##         :...credit_history in {critical,delayed}:
##             :...savings_balance = 101 - 500 DM: 1 (16.2/2.4)
##             :   savings_balance = < 100 DM:
##             :   :...other_debtors = co-applicant: 2 (7.5/2.4)
##             :       other_debtors = none:
##             :       :...personal_status = female: 1 (26.9/10.7)
##             :           personal_status in {married male,
##             :           :                   divorced male}: 2 (13.8/4)
##             :           personal_status = single male:
##             :           :...installment_rate <= 1: 1 (9.3)
##             :               installment_rate > 1:
##             :               :...credit_history = critical: 1 (38.8/10.7)
##             :                   credit_history = delayed: 2 (14.4/3.2)
##             credit_history in {repaid,fully repaid this bank,fully repaid}:
##             :...amount > 11054: 2 (16.9/0.8)
##                 amount <= 11054:
##                 :...job = mangement self-employed: 1 (36.5/13.6)
##                     job = unemployed non-resident: 2 (4.5)
##                     job in {skilled employee,unskilled resident}:
##                     :...installment_rate <= 2:
##                         :...dependents > 1: 2 (11.2/2.4)
##                         :   dependents <= 1:
##                         :   :...installment_rate <= 1: 2 (24.8/9.3)
##                         :       installment_rate > 1: 1 (42.6/14.4)
##                         installment_rate > 2:
##                         :...personal_status in {female,married male,
##                             :                   divorced male}: 2 (79.5/19.9)
##                             personal_status = single male:
##                             :...savings_balance = 101 - 500 DM: 1 (9.1/1.6)
##                                 savings_balance = < 100 DM:
##                                 :...months_loan_duration <= 11: 1 (9.9/2.2)
##                                     months_loan_duration > 11: 2 (59.6/13.8)
##
## -----  Trial 2:  -----
##
## Decision tree:
##
## foreign_worker = no: 1 (27.8/3.9)
## foreign_worker = yes:
## :...checking_balance in {< 0 DM,1 - 200 DM,> 200 DM}:
```

```
##     :...property = unknown/none:
##     :   :...housing in {own,rent}: 2 (31.8/5.2)
##     :   :   housing = for free:
##     :   :   :...dependents > 1: 2 (23.5/5.4)
##     :   :       dependents <= 1:
##     :   :       :...employment_length in {0 - 1 yrs,4 - 7 yrs,
##     :   :           :                     unemployed}: 1 (18.3/2.4)
##     :   :           employment_length in {1 - 4 yrs,> 7 yrs}:
##     :   :           :...savings_balance in {unknown,< 100 DM,501 - 1000 DM,
##     :   :               :                   > 1000 DM}: 2 (31.3/7.9)
##     :   :               savings_balance = 101 - 500 DM: 1 (4.5/0.7)
##     :   property in {building society savings,real estate,other}:
##     :   :...purpose in {car (used),business,repairs,retraining,
##     :       :           others}: 1 (81.6/25.7)
##     :       purpose in {education,domestic appliances}: 2 (28.2/10.3)
##     :       purpose = radio/tv:
##     :       :...months_loan_duration > 36: 2 (15.1/1.3)
##     :       :   months_loan_duration <= 36:
##     :       :   :...credit_history in {repaid,critical,fully repaid this bank,
##     :       :       :                  delayed}: 1 (112.3/35.6)
##     :       :       credit_history = fully repaid: 2 (4.1)
##     :       purpose = car (new):
##     :       :...savings_balance = > 1000 DM: 1 (4.8)
##     :       :   savings_balance in {unknown,< 100 DM,101 - 500 DM,
##     :       :   :                   501 - 1000 DM}:
##     :       :   :...installment_plan = bank: 2 (15.7/2.6)
##     :       :       installment_plan = stores: 1 (1.3/0.7)
##     :       :       installment_plan = none:
##     :       :       :...dependents > 1: 1 (15.8/5.3)
##     :       :           dependents <= 1:
##     :       :           :...installment_rate <= 1: 1 (13.3/5.2)
##     :       :               installment_rate > 1: 2 (67.9/19.4)
##     :       purpose = furniture:
##     :       :...installment_plan = stores: 1 (5.5)
##     :           installment_plan in {none,bank}:
##     :           :...other_debtors = guarantor: 1 (3.9)
##     :               other_debtors in {none,co-applicant}:
##     :               :...savings_balance in {unknown,> 1000 DM}: 1 (10.1/2.9)
##     :                   savings_balance in {101 - 500 DM,
##     :                   :                   501 - 1000 DM}: 2 (3.5)
##     :                   savings_balance = < 100 DM:
##     :                   :...amount <= 4473: 1 (66.2/30.1)
##     :                       amount > 4473: 2 (7)
##     checking_balance = unknown:
##     :...other_debtors = guarantor: 1 (3.9)
##         other_debtors = co-applicant: 2 (13.6/5.2)
##         other_debtors = none:
##         :...installment_plan = bank: 2 (50/21.1)
##             installment_plan in {none,stores}:
##             :...purpose in {radio/tv,car (used),domestic appliances,retraining,
##                 :           others}: 1 (101.9/8.4)
##                 purpose in {car (new),furniture,business,education,repairs}:
##                 :...amount > 7763: 2 (14.9/2)
##                     amount <= 7763:
```

```
##                     :...credit_history in {critical,
##                         :                  fully repaid this bank}: 1 (42.2/4.6)
##                         credit_history in {repaid,delayed,fully repaid}:
##                         :...savings_balance in {unknown,101 - 500 DM,
##                             :                   501 - 1000 DM}: 1 (28.8/8.4)
##                             savings_balance = > 1000 DM: 2 (8.2/2.6)
##                             savings_balance = < 100 DM:
##                             :...amount <= 1778: 1 (10.5)
##                                 amount > 1778: 2 (32.3/9.2)
##
## -----  Trial 3:  -----
##
## Decision tree:
##
## checking_balance in {unknown,> 200 DM}:
## :...foreign_worker = no: 1 (9.9)
## :   foreign_worker = yes:
## :   :...employment_length in {4 - 7 yrs,> 7 yrs}:
## :       :...dependents <= 1: 1 (112.3/11.8)
## :       :   dependents > 1:
## :       :   :...checking_balance = unknown: 1 (34.4/11.6)
## :       :       checking_balance = > 200 DM: 2 (5/0.5)
## :       employment_length in {1 - 4 yrs,0 - 1 yrs,unemployed}:
## :       :...other_debtors = guarantor: 1 (2.7)
## :           other_debtors = co-applicant: 2 (12.8/3.5)
## :           other_debtors = none:
## :           :...purpose in {car (used),education,domestic appliances,
## :               :           retraining,others}: 1 (36.8/9.2)
## :               purpose in {business,repairs}: 2 (33.3/9.6)
## :               purpose = car (new):
## :               :...housing in {own,for free}: 1 (28.7/9.2)
## :               :   housing = rent: 2 (5.6)
## :               purpose = furniture:
## :               :...job in {skilled employee,
## :               :   :       unemployed non-resident}: 1 (18.3/2.3)
## :               :   job in {mangement self-employed,
## :               :           unskilled resident}: 2 (18.6/6.9)
## :               purpose = radio/tv:
## :               :...job = unemployed non-resident: 1 (0)
## :                   job in {mangement self-employed,
## :                   :       unskilled resident}: 2 (16.8/5.6)
## :                   job = skilled employee:
## :                   :...amount <= 4057: 1 (28.2/1.5)
## :                       amount > 4057: 2 (9.1/3.5)
## checking_balance in {< 0 DM,1 - 200 DM}:
## :...credit_history in {fully repaid this bank,fully repaid}: 2 (64.1/18.1)
##     credit_history in {repaid,critical,delayed}:
##     :...other_debtors = guarantor: 1 (27.7/9.1)
##         other_debtors = co-applicant: 2 (23.3/9.6)
##         other_debtors = none:
##         :...purpose in {furniture,education,repairs,domestic appliances,
##             :           retraining}: 2 (134/54.6)
##             purpose in {business,others}: 1 (31/7.6)
##             purpose = car (used):
```

```
##             :...amount <= 8086: 1 (28.8/7.8)
##             :   amount > 8086: 2 (11.8/1.2)
##             purpose = car (new):
##             :...amount > 11054: 2 (6.5)
##             :   amount <= 11054:
##             :   :...personal_status = married male: 1 (5.7/2.6)
##             :       personal_status = divorced male: 2 (7.5/2.4)
##             :       personal_status = female:
##             :       :...amount <= 7418: 2 (26/5.7)
##             :       :   amount > 7418: 1 (4.4)
##             :       personal_status = single male:
##             :       :...months_loan_duration <= 42: 1 (55.8/16.6)
##             :           months_loan_duration > 42: 2 (2.9)
##             purpose = radio/tv:
##             :...foreign_worker = no: 1 (2.4)
##                 foreign_worker = yes:
##                 :...savings_balance in {unknown,501 - 1000 DM,
##                     :                   > 1000 DM}: 1 (22.5/7.8)
##                     savings_balance = 101 - 500 DM: 2 (10.8/1.1)
##                     savings_balance = < 100 DM:
##                     :...months_loan_duration > 39: 2 (6.1)
##                         months_loan_duration <= 39:
##                         :...amount > 3275: 1 (7.4)
##                             amount <= 3275:
##                             :...months_loan_duration <= 13: 1 (19.4/6.9)
##                                 months_loan_duration > 13: 2 (29/7.4)
##
## -----  Trial 4:  -----
##
## Decision tree:
##
## checking_balance in {unknown,> 200 DM}:
## :...purpose in {radio/tv,car (used),education,domestic appliances,retraining,
## :   :           others}: 1 (169.6/44.2)
## :   purpose = repairs: 2 (8.9/4.1)
## :   purpose = business:
## :   :...employment_length in {1 - 4 yrs,4 - 7 yrs,> 7 yrs}: 1 (29.9/6.4)
## :   :   employment_length in {0 - 1 yrs,unemployed}: 2 (12.5/1.1)
## :   purpose = car (new):
## :   :...installment_plan in {bank,stores}: 2 (20.3/7)
## :   :   installment_plan = none:
## :   :   :...amount <= 11760: 1 (51.5/14.6)
## :   :       amount > 11760: 2 (2.8)
## :   purpose = furniture:
## :   :...credit_history in {critical,fully repaid this bank,
## :       :                  fully repaid}: 1 (15.2)
## :       credit_history in {repaid,delayed}:
## :       :...other_debtors = guarantor: 1 (0)
## :           other_debtors = co-applicant: 2 (3.9)
## :           other_debtors = none:
## :           :...months_loan_duration <= 30: 1 (36.8/11.6)
## :               months_loan_duration > 30: 2 (4.4/0.5)
## checking_balance in {< 0 DM,1 - 200 DM}:
## :...savings_balance in {unknown,> 1000 DM}:
```

```
##     :...credit_history in {critical,delayed,fully repaid}: 1 (23.5/3.2)
##     :   credit_history = fully repaid this bank: 2 (5.2/2.1)
##     :   credit_history = repaid:
##     :   :...amount <= 5771: 1 (41.4/8.8)
##     :       amount > 5771: 2 (15/2.2)
##     savings_balance in {< 100 DM,101 - 500 DM,501 - 1000 DM}:
##     :...months_loan_duration > 42: 2 (37.3/8.8)
##         months_loan_duration <= 42:
##         :...purpose in {car (used),domestic appliances,retraining,
##             :           others}: 1 (47.4/17.5)
##             purpose in {education,repairs}: 2 (34.4/15)
##             purpose = business:
##             :...months_loan_duration <= 18: 1 (10.2)
##             :   months_loan_duration > 18: 2 (20.1/5.7)
##             purpose = car (new):
##             :...other_debtors in {guarantor,co-applicant}: 2 (15/3.5)
##             :   other_debtors = none:
##             :   :...installment_rate <= 3:
##             :       :...residence_history <= 1: 2 (5.9)
##             :       :   residence_history > 1: 1 (44.1/15.8)
##             :       installment_rate > 3:
##             :       :...amount <= 609: 1 (4.6)
##             :           amount > 609: 2 (37.7/7.8)
##             purpose = radio/tv:
##             :...foreign_worker = no: 1 (2.9)
##             :   foreign_worker = yes:
##             :   :...employment_length in {0 - 1 yrs,unemployed}: 2 (33.9/12.6)
##             :       employment_length in {4 - 7 yrs,> 7 yrs}: 1 (26.5/8.4)
##             :       employment_length = 1 - 4 yrs:
##             :       :...months_loan_duration <= 11: 1 (6.2)
##             :           months_loan_duration > 11: 2 (28.4/10.8)
##             purpose = furniture:
##             :...installment_plan = stores: 1 (4.9)
##                 installment_plan in {none,bank}:
##                 :...credit_history in {critical,fully repaid}: 1 (32.9/13.4)
##                     credit_history in {fully repaid this bank,
##                     :                  delayed}: 2 (8.1/1.1)
##                     credit_history = repaid:
##                     :...checking_balance = 1 - 200 DM: 2 (17.6/6)
##                         checking_balance = < 0 DM:
##                         :...months_loan_duration <= 15: 1 (16.5/2)
##                             months_loan_duration > 15: 2 (24.3/11)
##
## -----  Trial 5:  -----
##
## Decision tree:
##
## foreign_worker = no: 1 (23.7/4.5)
## foreign_worker = yes:
## :...checking_balance = < 0 DM:
##     :...job = mangement self-employed:
##     :   :...installment_rate <= 1: 2 (4.3)
##     :   :   installment_rate > 1: 1 (37.5/9.2)
##     :   job in {skilled employee,unskilled resident,unemployed non-resident}:
```

```
##     :   :...months_loan_duration <= 8: 1 (11.8/1.4)
##     :       months_loan_duration > 8:
##     :       :...purpose in {car (new),car (used),education,repairs,
##     :           :           domestic appliances,retraining}: 2 (102.3/33)
##     :           purpose in {furniture,business,others}: 1 (73.2/31.3)
##     :           purpose = radio/tv:
##     :           :...employment_length in {1 - 4 yrs,0 - 1 yrs,4 - 7 yrs,
##     :               :                     unemployed}: 2 (33.9/9.2)
##     :               employment_length = > 7 yrs: 1 (4)
##     checking_balance = > 200 DM:
##     :...dependents > 1: 2 (7.3/0.9)
##     :   dependents <= 1:
##     :   :...age > 39: 1 (14.2)
##     :       age <= 39:
##     :       :...age <= 24: 1 (7.4)
##     :           age > 24:
##     :           :...installment_plan in {none,stores}: 2 (31.1/8.4)
##     :               installment_plan = bank: 1 (3.7)
##     checking_balance = 1 - 200 DM:
##     :...employment_length = 4 - 7 yrs: 1 (42/8.4)
##     :   employment_length in {1 - 4 yrs,0 - 1 yrs,> 7 yrs,unemployed}:
##     :   :...amount > 12204: 2 (12.4)
##     :       amount <= 12204:
##     :       :...dependents > 1: 2 (24.6/7.7)
##     :           dependents <= 1:
##     :           :...housing = for free: 1 (20.7/4.2)
##     :               housing = rent:
##     :               :...savings_balance = 101 - 500 DM: 2 (11.1)
##     :               :   savings_balance in {unknown,< 100 DM,501 - 1000 DM,
##     :               :   :                   > 1000 DM}:
##     :               :   :...employment_length in {1 - 4 yrs,
##     :               :       :                     unemployed}: 1 (8.3)
##     :               :       employment_length in {0 - 1 yrs,
##     :               :                             > 7 yrs}: 2 (23.4/7.5)
##     :               housing = own:
##     :               :...residence_history <= 1: 1 (36.1/6.9)
##     :                   residence_history > 1:
##     :                   :...savings_balance = unknown: 1 (12.9/1)
##     :                       savings_balance in {< 100 DM,101 - 500 DM,
##     :                       :                   501 - 1000 DM,> 1000 DM}:
##     :                       :...job in {mangement self-employed,
##     :                           :       unemployed non-resident}: 2 (13.4/1.7)
##     :                           job in {skilled employee,unskilled resident}:
##     :                           :...employment_length = 0 - 1 yrs: 2 (17.4/6.5)
##     :                               employment_length in {> 7 yrs,
##     :                               :                     unemployed}: 1 (14.3/2.9)
##     :                               employment_length = 1 - 4 yrs:
##     :                               :...months_loan_duration > 22: 2 (7.2/0.4)
##     :                                   months_loan_duration <= 22:
##     :                                   :...age <= 55: 1 (26.1/5.7)
##     :                                       age > 55: 2 (5.6/0.8)
##     checking_balance = unknown:
##     :...credit_history in {critical,fully repaid this bank}: 1 (97.7/24.4)
##         credit_history = fully repaid: 2 (7/3.5)
##         credit_history = delayed:
##         :...installment_rate <= 3: 1 (14.3/2.7)
##         :   installment_rate > 3: 2 (23.2/5.4)
##         credit_history = repaid:
##         :...savings_balance = 101 - 500 DM: 1 (9.8)
##             savings_balance in {unknown,< 100 DM,501 - 1000 DM,> 1000 DM}:
##             :...existing_credits > 1: 2 (24.7/8.8)
##                 existing_credits <= 1:
##                 :...age > 41: 1 (14.8/1.6)
##                     age <= 41:
##                     :...residence_history <= 1: 1 (4.3)
##                         residence_history > 1:
##                         :...savings_balance in {unknown,
##                             :                   > 1000 DM}: 1 (16.1/1.8)
##                             savings_balance in {< 100 DM,501 - 1000 DM}:
##                             :...personal_status in {married male,
##                                 :                   divorced male}: 1 (3.8)
##                                 personal_status in {female,single male}:
##                                 :...telephone = yes: 1 (13.8/5.3)
##                                     telephone = none: [S1]
##
## SubTree [S1]
##
## job = unemployed non-resident: 2 (0)
## job = mangement self-employed: 1 (2.4)
## job in {skilled employee,unskilled resident}:
## :...months_loan_duration <= 30: 2 (35.3/7.8)
##     months_loan_duration > 30: 1 (2.9)
##
## -----  Trial 6:  -----
##
## Decision tree:
##
## amount > 6419: 2 (134.8/52.8)
## amount <= 6419:
## :...months_loan_duration <= 7: 1 (54.4/10.7)
##     months_loan_duration > 7:
##     :...checking_balance = unknown:
##         :...installment_plan = stores: 1 (14.1/6.9)
##         :   installment_plan = bank:
##         :   :...age > 43: 1 (8.3)
##         :   :   age <= 43:
##         :   :   :...age <= 31: 1 (11.9/2.9)
##         :   :       age > 31: 2 (20.4/3.4)
##         :   installment_plan = none:
##         :   :...credit_history in {critical,fully repaid this bank,
##         :       :                  fully repaid}: 1 (43.1)
##         :       credit_history in {repaid,delayed}:
##         :       :...residence_history <= 1: 2 (10.4/2.6)
##         :           residence_history > 1:
##         :           :...savings_balance in {unknown,101 - 500 DM}: 1 (19.9)
##         :               savings_balance in {< 100 DM,501 - 1000 DM,> 1000 DM}:
##         :               :...other_debtors = guarantor: 1 (0.3)
##         :                   other_debtors = co-applicant: 2 (4.1/0.7)
##         :                   other_debtors = none:
##         :                   :...age > 29: 1 (35.2/6.9)
##         :                       age <= 29:
##         :                       :...installment_rate <= 3: 1 (17.4/5.7)
##         :                           installment_rate > 3: 2 (18.5/3.6)
##         checking_balance in {< 0 DM,1 - 200 DM,> 200 DM}:
##         :...residence_history <= 1:
##             :...installment_plan in {bank,stores}: 1 (16.5)
##             :   installment_plan = none:
##             :   :...other_debtors in {guarantor,co-applicant}: 1 (3.5)
##             :       other_debtors = none:
##             :       :...job = mangement self-employed: 1 (5)
##             :           job in {unskilled resident,
##             :           :       unemployed non-resident}: 2 (9.7/0.9)
##             :           job = skilled employee:
##             :           :...housing in {rent,for free}: 2 (7.8/1.2)
##             :               housing = own:
##             :               :...checking_balance = < 0 DM: 2 (8.8/1.9)
##             :                   checking_balance in {1 - 200 DM,
##             :                                        > 200 DM}: 1 (32.2/5.5)
##             residence_history > 1:
##             :...installment_rate > 2:
##                 :...job = unemployed non-resident: 1 (5.4)
##                 :   job in {skilled employee,mangement self-employed,
##                 :   :       unskilled resident}:
##                 :   :...telephone = none:
##                 :       :...installment_plan in {bank,stores}: 2 (40.7/7.7)
##                 :       :   installment_plan = none:
##                 :       :   :...personal_status = female: 2 (44.5/13.9)
##                 :       :       personal_status in {married male,
##                 :       :       :                   divorced male}: 1 (20.3/8.8)
##                 :       :       personal_status = single male: [S1]
##                 :       telephone = yes:
##                 :       :...other_debtors in {guarantor,
##                 :           :                 co-applicant}: 1 (12.3/0.9)
##                 :           other_debtors = none:
##                 :           :...savings_balance in {unknown,101 - 500 DM,
##                 :               :                   > 1000 DM}: 1 (24.5/5.1)
##                 :               savings_balance = 501 - 1000 DM: 2 (6.3/2.7)
##                 :               savings_balance = < 100 DM: [S2]
##                 installment_rate <= 2:
##                 :...credit_history = delayed: 1 (9)
##                     credit_history in {repaid,critical,fully repaid this bank,
##                     :                  fully repaid}:
##                     :...housing in {rent,for free}: 1 (49.7/15.4)
##                         housing = own:
##                         :...foreign_worker = no: 2 (2.6/0.3)
##                             foreign_worker = yes:
##                             :...installment_rate <= 1: 1 (15.2/4.8)
##                                 installment_rate > 1:
##                                 :...months_loan_duration > 36: 2 (5)
##                                     months_loan_duration <= 36: [S3]
##
## SubTree [S1]
```

```
##
## savings_balance in {unknown,101 - 500 DM}: 1 (18.4/3.8)
## savings_balance in {501 - 1000 DM,> 1000 DM}: 2 (5.9/0.7)
## savings_balance = < 100 DM:
## :...property in {building society savings,real estate}: 1 (37.6/13)
##     property in {other,unknown/none}: 2 (22.4/4.7)
##
## SubTree [S2]
##
## credit_history in {fully repaid this bank,delayed,fully repaid}: 2 (11.2)
## credit_history in {repaid,critical}:
## :...job in {skilled employee,unskilled resident}: 2 (28.2/9.4)
##     job = mangement self-employed: 1 (10.6/2)
##
## SubTree [S3]
##
## other_debtors = guarantor: 1 (2.7)
## other_debtors = co-applicant: 2 (4.6/1.9)
## other_debtors = none:
## :...amount <= 3416: 2 (33.8/14.4)
##     amount > 3416: 1 (11.6/1.3)
##
## -----  Trial 7:  -----
##
## Decision tree:
##
## credit_history in {fully repaid this bank,fully repaid}:
## :...property in {building society savings,unknown/none}: 2 (47.1/12.3)
## :   property in {real estate,other}:
## :   :...savings_balance in {unknown,< 100 DM}: 2 (41.9/17.7)
## :       savings_balance in {101 - 500 DM,501 - 1000 DM,> 1000 DM}: 1 (15.2/1.3)
## credit_history in {repaid,critical,delayed}:
## :...checking_balance in {unknown,> 200 DM}: 1 (304.8/95)
##     checking_balance in {< 0 DM,1 - 200 DM}:
##     :...property = real estate:
##         :...savings_balance in {unknown,101 - 500 DM,501 - 1000 DM,
##         :   :                   > 1000 DM}: 1 (28.1/3.9)
##         :   savings_balance = < 100 DM:
##         :   :...age > 33: 1 (43.4/7.3)
##         :       age <= 33:
##         :       :...amount <= 1217: 1 (17.7/3.7)
##         :           amount > 1217: 2 (33/7.9)
##         property in {building society savings,other,unknown/none}:
##         :...amount <= 959: 2 (37.7/8.2)
##             amount > 959:
##             :...dependents > 1: 1 (50/16.5)
##                 dependents <= 1:
##                 :...months_loan_duration > 27:
##                     :...job = unskilled resident: 2 (6.5)
##                     :   job = unemployed non-resident: 1 (2.7)
##                     :   job in {skilled employee,mangement self-employed}:
##                     :   :...credit_history = delayed: 1 (17.1/7.4)
##                     :       credit_history in {repaid,critical}:
##                     :       :...residence_history <= 1: 1 (6.3/1.1)
##                     :           residence_history > 1: 2 (49.2/14.5)
##                     months_loan_duration <= 27:
##                     :...personal_status = married male: 2 (16.7/4.4)
##                         personal_status in {female,single male,divorced male}:
##                         :...credit_history in {critical,delayed}: 1 (64.6/13.5)
##                             credit_history = repaid:
##                             :...amount > 10222: 2 (5.8)
##                                 amount <= 10222:
##                                 :...age > 54: 1 (10.3/1.3)
##                                     age <= 54:
##                                     :...age <= 31: 1 (66.7/19.4)
##                                         age > 31: 2 (32/8.5)
##
## -----  Trial 8:  -----
##
## Decision tree:
##
## housing in {rent,for free}:
## :...purpose in {business,repairs,domestic appliances,retraining}: 2 (31.2/6.8)
## :   purpose in {education,others}: 1 (30.1/11.4)
## :   purpose = car (used):
## :   :...amount <= 11054: 1 (40.8/12.4)
## :   :   amount > 11054: 2 (5.8)
## :   purpose = car (new):
## :   :...employment_length = unemployed: 1 (8.8/0.7)
## :   :   employment_length in {1 - 4 yrs,0 - 1 yrs,4 - 7 yrs,> 7 yrs}:
## :   :   :...months_loan_duration <= 9: 1 (3.9)
## :   :       months_loan_duration > 9: 2 (59.7/16.1)
## :   purpose = furniture:
## :   :...credit_history = delayed: 1 (0)
## :   :   credit_history in {fully repaid this bank,fully repaid}: 2 (6.1)
## :   :   credit_history in {repaid,critical}:
## :   :   :...job in {skilled employee,unskilled resident,
## :   :       :       unemployed non-resident}: 1 (45.9/15.3)
## :   :       job = mangement self-employed: 2 (12.6/3.2)
## :   purpose = radio/tv:
## :   :...job in {mangement self-employed,unemployed non-resident}: 1 (10.8)
## :       job in {skilled employee,unskilled resident}:
## :       :...employment_length in {1 - 4 yrs,0 - 1 yrs,4 - 7 yrs,
## :           :                     unemployed}: 2 (28.6/8)
## :           employment_length = > 7 yrs: 1 (3.9)
## housing = own:
## :...purpose in {car (used),repairs,domestic appliances,
##     :           retraining}: 1 (72.6/14.3)
##     purpose in {education,others}: 2 (27.1/11.7)
##     purpose = car (new):
##     :...foreign_worker = no: 1 (5.2)
##     :   foreign_worker = yes:
##     :   :...installment_rate <= 2:
##     :       :...existing_credits > 3: 2 (2.2)
##     :       :   existing_credits <= 3:
##     :       :   :...age <= 23: 2 (3.9)
##     :       :       age > 23: 1 (42.6/10.5)
##     :       installment_rate > 2:
```

```
##     :       :...installment_plan in {bank,stores}: 2 (13.1)
##     :           installment_plan = none:
##     :           :...checking_balance in {< 0 DM,1 - 200 DM,
##     :               :                    > 200 DM}: 2 (57.9/24.1)
##     :               checking_balance = unknown: 1 (8.7)
##     purpose = furniture:
##     :...installment_plan = stores: 1 (9)
##     :   installment_plan in {none,bank}:
##     :   :...credit_history = fully repaid this bank: 2 (4.1)
##     :       credit_history = fully repaid: 1 (5.3)
##     :       credit_history in {repaid,critical,delayed}:
##     :       :...telephone = none:
##     :           :...months_loan_duration <= 15: 1 (26.7/7.9)
##     :           :   months_loan_duration > 15: 2 (42.8/12.8)
##     :           telephone = yes:
##     :           :...job in {skilled employee,unskilled resident,
##     :               :       unemployed non-resident}: 1 (19.9/1.2)
##     :               job = mangement self-employed: 2 (11.9/4.7)
##     purpose = radio/tv:
##     :...checking_balance = unknown: 1 (49.7/5.9)
##     :   checking_balance in {< 0 DM,1 - 200 DM,> 200 DM}:
##     :   :...months_loan_duration > 36: 2 (13.4/1.5)
##     :       months_loan_duration <= 36:
##     :       :...other_debtors = guarantor: 1 (12.1)
##     :           other_debtors = co-applicant: 2 (2.3)
##     :           other_debtors = none:
##     :           :...employment_length in {1 - 4 yrs,4 - 7 yrs,
##     :               :                     > 7 yrs}: 1 (61.4/17.7)
##     :               employment_length in {0 - 1 yrs,unemployed}: 2 (29.2/8.7)
##     purpose = business:
##     :...savings_balance in {unknown,101 - 500 DM,501 - 1000 DM}: 1 (23.1/4)
##         savings_balance = > 1000 DM: 2 (6.4/3.1)
##         savings_balance = < 100 DM:
##         :...amount > 7596: 2 (6.3)
##             amount <= 7596:
##             :...installment_plan = bank: 1 (5.7)
##                 installment_plan in {none,stores}:
##                 :...telephone = none: 1 (16.9/4.3)
##                     telephone = yes: 2 (23.1/8.5)
##
## -----  Trial 9:  -----
##
## Decision tree:
##
## checking_balance = unknown:
## :...employment_length in {4 - 7 yrs,> 7 yrs}: 1 (89.1/5.2)
## :   employment_length in {1 - 4 yrs,0 - 1 yrs,unemployed}:
## :   :...installment_plan in {bank,stores}:
## :       :...other_debtors in {guarantor,co-applicant}: 1 (3.9)
## :       :   other_debtors = none:
## :       :   :...residence_history <= 1: 1 (3.2)
## :       :       residence_history > 1:
## :       :       :...purpose in {car (new),furniture,car (used),business,
## :       :           :           repairs,domestic appliances,retraining,
## :       :           :           others}: 2 (30.3/4.6)
## :       :           purpose in {radio/tv,education}: 1 (4.7)
## :       installment_plan = none:
## :       :...other_debtors = co-applicant: 2 (7.9/1.4)
## :           other_debtors in {none,guarantor}:
## :           :...months_loan_duration <= 16: 1 (31)
## :               months_loan_duration > 16:
## :               :...property in {building society savings,
## :                   :            unknown/none}: 1 (15.6)
## :                   property in {real estate,other}:
## :                   :...credit_history in {repaid,delayed}: 2 (27.6/10.8)
## :                       credit_history in {critical,fully repaid this bank,
## :                                          fully repaid}: 1 (9.7)
## checking_balance in {< 0 DM,1 - 200 DM,> 200 DM}:
## :...savings_balance in {unknown,501 - 1000 DM,> 1000 DM}:
##     :...savings_balance = > 1000 DM: 1 (26.7/6.9)
##     :   savings_balance in {unknown,501 - 1000 DM}:
##     :   :...installment_rate > 3:
##     :       :...residence_history <= 3: 2 (30.5/9.5)
##     :       :   residence_history > 3: 1 (24.2/2.5)
##     :       installment_rate <= 3:
##     :       :...housing = rent: 1 (8.7)
##     :           housing = for free: 2 (5.9/1.6)
##     :           housing = own:
##     :           :...age <= 23: 2 (4.8)
##     :               age > 23: 1 (30.2/1.8)
##     savings_balance in {< 100 DM,101 - 500 DM}:
##     :...months_loan_duration > 47: 2 (31.5/6.1)
##         months_loan_duration <= 47:
##         :...other_debtors = co-applicant: 2 (27.6/13.5)
##             other_debtors = guarantor:
##             :...installment_plan in {none,stores}: 1 (20.5/3)
##             :   installment_plan = bank: 2 (11.7/4.3)
##             other_debtors = none:
##             :...credit_history in {fully repaid this bank,
##                 :                  fully repaid}: 2 (51.4/17.4)
##                 credit_history = delayed:
##                 :...installment_rate <= 1: 1 (6.1)
##                 :   installment_rate > 1:
##                 :   :...savings_balance = < 100 DM: 2 (23/8)
##                 :       savings_balance = 101 - 500 DM: 1 (10.3/2.9)
##                 credit_history = critical:
##                 :...savings_balance = 101 - 500 DM: 1 (7.4/1)
##                 :   savings_balance = < 100 DM:
##                 :   :...personal_status = divorced male: 2 (9/1.3)
##                 :       personal_status in {female,single male,married male}:
##                 :       :...telephone = yes: 1 (31.3/6.8)
##                 :           telephone = none: [S1]
##                 credit_history = repaid:
##                 :...installment_rate <= 1: 2 (31/9.9)
##                     installment_rate > 1:
##                     :...job = unemployed non-resident: 2 (1.5)
##                         job = mangement self-employed:
##                         :...amount <= 7582: 1 (26.9/3.8)
##                          :   amount > 7582: 2 (4.2)
##                          job in {skilled employee,unskilled resident}:
##                          :...foreign_worker = no: 1 (2.2)
##                              foreign_worker = yes:
##                              :...installment_plan = stores: 1 (4.4)
##                                  installment_plan in {none,bank}:
##                                  :...installment_rate > 3: 2 (85.2/32.7)
##                                      installment_rate <= 3: [S2]
##
## SubTree [S1]
##
## property in {building society savings,real estate,unknown/none}: 1 (37.7/10)
## property = other: 2 (7.9/0.2)
##
## SubTree [S2]
##
## personal_status in {single male,married male}: 1 (50.5/13.1)
## personal_status in {female,divorced male}:
## :...existing_credits > 1: 2 (3.9)
##     existing_credits <= 1:
##     :...savings_balance = < 100 DM: 2 (36.5/13.2)
##         savings_balance = 101 - 500 DM: 1 (3.2)
##
##
## Evaluation on training data (900 cases):
##
## Trial        Decision Tree
## -----      ----------------
##    Size      Errors
##
##    0     42  136(15.1%)
##    1     27  189(21.0%)
##    2     33  215(23.9%)
##    3     36  210(23.3%)
##    4     36  180(20.0%)
##    5     42  199(22.1%)
##    6     44  216(24.0%)
##    7     21  190(21.1%)
##    8     41  211(23.4%)
##    9     41  181(20.1%)
## boost             45( 5.0%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     628     7    (a): class 1
##      38   227    (b): class 2
##
##
##  Attribute usage:
##
##  100.00% checking_balance
##  100.00% credit_history
##  100.00% purpose
##  100.00% amount
##  100.00% housing
##  100.00% foreign_worker
##   99.56% other_debtors
##   96.78% months_loan_duration
##   87.56% employment_length
##   85.89% installment_plan
##   83.89% savings_balance
##   74.22% residence_history
##   70.11% dependents
##   70.00% property
##   68.56% installment_rate
##   68.00% job
##   51.33% personal_status
##   49.44% age
##   47.00% telephone
##   23.22% existing_credits
##
##
## Time: 0.0 secs
```

```
# make prediction
credit_boost_pred10 <- predict(credit_boost10,
                               credit_test)
```

7. Evaluate the boosted prediction

```
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 |         1 |         2 | Row Total |
## ---------------|-----------|-----------|-----------|
##              1 |        59 |         6 |        65 |
##                |     0.590 |     0.060 |           |
## ---------------|-----------|-----------|-----------|
##              2 |        17 |        18 |        35 |
##                 |      0.170 |      0.180 |            |
## ----------------|------------|------------|------------|
##    Column Total |         76 |         24 |        100 |
## ----------------|------------|------------|------------|
## 
## 
```

As the table shows, the accuracy is (59 + 18) / 100 = 0.77 now, improved from before.

8. Making mistakes more costlier than others

```
# create matrix of different predictions
matrix_dimensions <- list(c("1", "2"),
                          c("1", "2"))
names(matrix_dimensions) <- c("predicted",
                              "actual")
matrix_dimensions
```

```
## $predicted
## [1] "1" "2"
## 
## $actual
## [1] "1" "2"
```

```
# give values to the matrix for penalty
error_cost <- matrix(c(0, 1, 4, 0),
                     nrow = 2,
                     dimnames = matrix_dimensions)
error_cost
```

```
##          actual
## predicted 1 2
##         1 0 4
##         2 1 0
```

9. Make predictions based on the penalty matrix

```
# apply cost to decision tree using costs prameter
credit_cost <- C5.0(credit_train[-17],
                    credit_train$default,
                    costs = error_cost)
#summary(credit_cost)
# make prediction
credit_cost_pred <- predict(credit_cost,
                            credit_test)
# evaluate the prediction
CrossTable(credit_test$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 |         1 |         2 | Row Total |
## ---------------|-----------|-----------|-----------|
##              1 |        43 |        22 |        65 |
##                |     0.430 |     0.220 |           |
## ---------------|-----------|-----------|-----------|
##              2 |         8 |        27 |        35 |
##                |     0.080 |     0.270 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |        51 |        49 |       100 |
## ---------------|-----------|-----------|-----------|
##
##
```

As the table shows, the accuracy is (43 + 27) / 100 = 0.7. The overall accuracy dropped a bit. Compare to the boosted model, the false negative prediction increased and false positive decreased as we are giving false negative a higher cost.

Problem 2

1. Read in data

```
mushrooms <- read.csv("mushrooms.csv", stringsAsFactors = TRUE)
```

2. Get to know the data

```
# check the structure
str(mushrooms)
```

```
## 'data.frame':    8124 obs. of  23 variables:
##  $ type                    : Factor w/ 2 levels "edible","poisonous": 2 1 1 2 1 1 1 1 2 1 ...
##  $ cap_shape               : Factor w/ 6 levels "bell","conical",..: 3 3 1 3 3 3 1 1 3 1 ...
##  $ cap_surface             : Factor w/ 4 levels "fibrous","grooves",..: 4 4 4 3 4 3 4 3 3 4 ...
##  $ cap_color               : Factor w/ 10 levels "brown","buff",..: 1 10 9 9 4 10 9 9 9 10 ...
##  $ bruises                 : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
##  $ odor                    : Factor w/ 9 levels "almond","anise",..: 8 1 2 8 7 1 1 2 8 1 ...
##  $ gill_attachment         : Factor w/ 2 levels "attached","free": 2 2 2 2 2 2 2 2 2 2 ...
##  $ gill_spacing            : Factor w/ 2 levels "close","crowded": 1 1 1 1 2 1 1 1 1 1 ...
##  $ gill_size               : Factor w/ 2 levels "broad","narrow": 2 1 1 2 1 1 1 1 2 1 ...
##  $ gill_color              : Factor w/ 12 levels "black","brown",..: 1 1 2 2 1 2 5 2 8 5 ...
##  $ stalk_shape             : Factor w/ 2 levels "enlarging","tapering": 1 1 1 1 2 1 1 1 1 1 ...
##  $ stalk_root              : Factor w/ 5 levels "bulbous","club",..: 3 2 2 3 3 2 2 2 3 2 ...
##  $ stalk_surface_above_ring: Factor w/ 4 levels "fibrous","scaly",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ stalk_surface_below_ring: Factor w/ 4 levels "fibrous","scaly",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ stalk_color_above_ring  : Factor w/ 9 levels "brown","buff",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ stalk_color_below_ring  : Factor w/ 9 levels "brown","buff",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ veil_type               : Factor w/ 1 level "partial": 1 1 1 1 1 1 1 1 1 1 ...
##  $ veil_color              : Factor w/ 4 levels "brown","orange",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ ring_number             : Factor w/ 3 levels "none","one","two": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ring_type               : Factor w/ 5 levels "evanescent","flaring",..: 5 5 5 5 1 5 5 5 5 5 ...
##  $ spore_print_color       : Factor w/ 9 levels "black","brown",..: 1 2 2 1 2 1 1 2 1 1 ...
##  $ population              : Factor w/ 6 levels "abundant","clustered",..: 4 3 3 4 1 3 3 4 5 4 ...
##  $ habitat                 : Factor w/ 7 levels "grasses","leaves",..: 5 1 3 5 1 1 3 3 1 3 ...
```

```r
# drop the veil type column
mushrooms$veil_type <- NULL
# check the proportion of mushroom types
table(mushrooms$type)
```

```
##
##    edible poisonous
##      4208      3916
```

3. Train the model

```r
# train the data using all features
mushroom_1R <- OneR(type ~., data = mushrooms)
mushroom_1R
```

```
## odor:
##  almond  -> edible
##  anise   -> edible
##  creosote    -> poisonous
##  fishy   -> poisonous
##  foul    -> poisonous
##  musty   -> poisonous
##  none    -> edible
##  pungent -> poisonous
##  spicy   -> poisonous
## (8004/8124 instances correct)
```

4. Evaluate model performance

```r
summary(mushroom_1R)
```

```
##
## === Summary ===
##
## Correctly Classified Instances        8004               98.5229 %
## Incorrectly Classified Instances       120                1.4771 %
## Kappa statistic                          0.9704
## Mean absolute error                      0.0148
## Root mean squared error                  0.1215
## Relative absolute error                  2.958  %
## Root relative squared error             24.323  %
## Total Number of Instances             8124
##
## === Confusion Matrix ===
##
##     a    b   <-- classified as
##  4208    0 |    a = edible
##   120 3796 |    b = poisonous
```

As the matrix shows, there are 120 posionous mushrooms were classified as edible.

5. Improve model performance

```r
# apply RIPPEr rule to predict type
mushroom_JRip <- JRip(type ~ ., data = mushrooms)
mushroom_JRip
```

```
## JRIP rules:
## ===========
##
## (odor = foul) => type=poisonous (2160.0/0.0)
## (gill_size = narrow) and (gill_color = buff) => type=poisonous (1152.0/0.0)
## (gill_size = narrow) and (odor = pungent) => type=poisonous (256.0/0.0)
## (odor = creosote) => type=poisonous (192.0/0.0)
## (spore_print_color = green) => type=poisonous (72.0/0.0)
## (stalk_surface_below_ring = scaly) and (stalk_surface_above_ring = silky) => type=poisonous (68.0/0.
## (habitat = leaves) and (cap_color = white) => type=poisonous (8.0/0.0)
## (stalk_color_above_ring = yellow) => type=poisonous (8.0/0.0)
##  => type=edible (4208.0/0.0)
##
## Number of Rules : 9
```

6. Evaluate RIPPER Rule learner

```r
summary(mushroom_JRip)
```

```
##
## === Summary ===
##
## Correctly Classified Instances        8124              100      %
## Incorrectly Classified Instances         0                0      %
## Kappa statistic                          1
## Mean absolute error                      0
## Root mean squared error                  0
## Relative absolute error                  0      %
## Root relative squared error              0      %
## Total Number of Instances             8124
##
```

```
## === Confusion Matrix ===
##
##     a    b   <-- classified as
##  4208    0 |    a = edible
##     0 3916 |    b = poisonous
```

As the matrix, all mushrooms were classified correctly.

Problem 3

k-NN

k-NN is a non-parametric method, it is a type of instance-based learning or lazy learning. It relies on distance for classification. When using k-NN as a classifier, it first calculates the distance between test data and each row of training data. Then based on the distance value, choose the top k rows from the sorted array, k is a user-defined value. Then it will assign a class to the test point based on the most frequent class of these rows. Given that the distance formula is dependent on how features are measured, it usually requires transformations of features to a standard range before applying the k-NN algorithm. Min-max normalization and z-score standardization are commonly used.

k-NN is simple and effective. It makes no assumptions about the underlying data distribution and the training phase is fast. Therefore, it is great for dealing with "real world" data that most of which are not obeying the typical theoretical assumptions (e.g., linear regression), which also means very little or no prior knowledge about the distribution data is required. Based on these features, applications of k-NN could be: predict people's credit rating based on exisits credit rating database since people who have similar financial details would have similar credit ratings, or, classing a potential voter to "will vote"/"will not vote", or to "vote Democrat"/"vote Republican".

However, k-NN is not ideal when: 1) If the dataset is big and the speed matters. Since it needs to calculate the distance for each data point; 2) If the dataset has a lot of missing data. It cannot define the distance if one or more attributes are missing; 3) if the dataset needs a clear interpretation or deeper understanding. It does not produce a model. Also as it is non-parametric, it has limited ability to understand how the features are related to the class, e.g. "why is this data point classified under this class?", or "what is the relationship between this attribute and the class distribution?". Moreover, it relies heavily on the selection of an appropriate k.

Naive Bayes

The fundamental principle of Naive Bayes is Bayesian methods. Bayesian probability theory is based on the idea that the estimated likelihood of an event, or a potential outcome, should be based on the evidence at hand across multiple trials, or opportunities for the event to occur. Classifiers based on Bayesian methods such as Naive Bayes classifies an object by mapping its features with classifier individually then calculate the posterior probability to determine whether they are more likely to be.

Given its independence feature, it is typically applied to problems in which the information from numerous attributes should be considered simultaneously and it requires less training data and processing time. When the assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression and you need less training data. Therefore, it is great for real-time prediction since it is sure fast; it is great for text classification/spam filtering and recommendation systems. Moreover, it is one of the classifiers that handle missing data very well: it simply excludes the attribute with missing data when computing posterior probability (i.e. probability of class given data point).

However, Naive Bayes is not ideal when: 1) There is no conditional independence. It assumes that all of the features in the dataset are equally important and independent. Not having independence in the data will have a highly negative influence on classification; 2) If the data's decision boundary is nonlinear/elliptic/parabolic. For the same independency reason and its assumptions on all the numeric attributes are normally distributed, it can only have linear, elliptic, or parabolic decision boundaries; 3) There is a lot of zero frequency. It will result in any new data point which contains a zero-frequency attribute will be always classified as the other class. This can be solved by using Laplace estimator, which ensures that each feature has a nonzero probability of occurring with each class; 4) Numerical data. It is not ideal for datasets with many numeric features, it estimated probabilities are less reliable than the predicted classes. For numerical variable, normal distribution is assumed.

C5.0 Decision Tree

The decision tree utilizes a tree structure to model the relationships among the features and the potential outcomes. It is not necessarily exclusively for the learner's internal use. Entropy and information gain are two of the most important components that algorithms are using to make decisions of the splitting point. Entropy quantifies the randomness within a set of class values and information gain is the basic criterion to decide whether a feature should be used as a node to be split. The higher the information gain, the better a feature is at creating homogeneous groups after a split on this feature.

C5.0 algorithm is the most well-known implementation. After the model is created, many decision tree algorithms output the resulting structure in a human-readable format, which could provide insight into how and why the model works or doesn't work and it is also helpful to keep the transparency of the mechanism. It is an all-purpose classifier that does well on most problems, and it is a highly automatic learning process, which can handle numeric or nominal features, as well as missing data, meaning it requires less effort for data preparation. It excludes unimportant features and can be used on both small and large datasets, it results in a model that is intuitive and can be easily interpreted. It is more efficient than other complex models.

However, decision trees don't work well when: 1) There is a lot of uncorrelated variables in the data. Decision trees work by finding the interactions between variables; 2) The data has smooth boundaries. Decision trees work best when the data has a discontinuous piecewise-constant model, it doesn't work well with a linear target function. 3) When speed matters. Decision trees sometimes can go far more complex compared to other algorithms therefore they always involve higher time to train the model. They are relatively expensive as the complexity and time taken together. More importantly: 1) They are often biased toward splits on features having a large number of levels since each split in a tree leads to a reduced dataset; 2) It is easy to overfit or underfit the model; 3) Small changes in the training data can result in large changes to decision logic; 4) They can also have troubles modeling some relationships due to their axis-parallel splits.

RIPPER Rules

Classification rules represent knowledge in an if-else form logical rule that assigns a class to unlabeled examples, they are specified in terms of 1) antecedent, which comprises certain combinations of feature values, and 2) consequent, which specifies the class value to assign when the rule's conditions are met.

Rule learning algorithms start by finding rules that cover a subset of data, then separating additional subsets of the data until the entire dataset has been covered and no more examples remain. This process is usually slow when dealing with a large dataset, and they often prone to being inaccurate on noisy data. Incremental Reduced Error Pruning (IREP) was the first one tried to solve this problem by using a combination of pre-pruning and post-pruning methods that grow very complex rules and prune them before separating the instances from the full data set. RIPPER stands for Repeated Incremental Pruning to Produce Error Reduction which further improved upon IREP to generate rules that match or exceed the performance of decision trees. It first uses the separate and conquers technique to greedily add conditiions to a rule until it perfectly classifies a subset of data or runs out of attributes for splitting. Then the tree is pruned. The grow-prune cycle will be repeated until it reaches a stopping criterion, after which the entire set of the rules is optimized using a variety of heuristics.

RIPPER Rule learners offer some advantages over trees for some tasks. 1) Rules are easy to interpret. A decision tree must be applied from top-to-bottom through a series of decisions, rules are propositions that can be read much like a statement of fact. 2) The predictions are fast. Since only a few binary statements need to be checked to determine which rules to apply; 3) Rules usually generate sparse models. They only select the relevant features for the model. Rule learners are generally applied to problems where the features are primarily or entirely nominal. They perform very well at idenrifying rare events, even if the rare event occurs only for a very specific interaction among feature values.

However, RIPPER Rule learners also have their limitations that 1)It may result in rules that seem to defy common sense or expert knowledge; 2) It is not ideal for working with numeric data. Numeric features must be categorized; 3) It might not perform as well as more complex models.

Problem 4

Ensemble methods combine the decisions from multiple models to improve the overall performance. The main causes of error in learning models are due to noise, bias, and variance. Ensemble methods help to minimize these factors. These methods are designed to improve the stability and accuracy of machine learning algorithms. As long as the base models are diverse and independent, the prediction error of the model decreases when the ensemble approach is used. Even though the ensemble model has multiple base models within the model, it acts and performs as a single model.

Ensemble methods are used in almost all machine learning scenarios to enhance the prediction abilities of the models. It provides more accurate prediction results and it provides a stable and more robust model as the aggregate result of multiple models is always less noisy than the individual models. However, it will also reduce the model interpret-ability due to increased complexity and makes it very difficult to draw any crucial business insights at the end. The computation and design time is also high, which is not good for real-time applications. The results can be also affected by the selections of models.

Simple ensemble technique first takes the most frequently occurring number found in a set of numbers, then average or weighted average of the predictions from all the models and use it to make the final prediction.

Advanced ensemble techniques include bagging and boosting techniques. Bagging method first creates random samples of the training data with replacement, then build a model for each sample. Finally, the results of these multiple models are combined using average or majority voting. Since each model is exposed to a different subset, the combinations of their output will make sure that the problem of overfitting is taken care of by not clinging too closely to our training data set, meaning Bagging is helpful to reduce the variance error.

Boosting is sequential, the first algorithm is trained on the entire data set and the subsequent algorithms are built by fitting the residuals of the first algorithm, thus giving higher weight to those observations that were poorly predicted by the previous model. Therefore, it is actually creating a series of weak learners each of which might not be good for the entire data set but it is good for some part of the data set, each model will boost the performance of the ensemble. In gereral decreases the bias error and builds strong predictive models. It has shown better predictive accuracy than bagging but it also tends to overfit the data on the other hand.

Compare Bagging and Boosting, they both use voting and combines models of the same type, but bagging uses individual models that are built separately and they are given equal weight, while in Boosting, each new model is influenced by the performance of those built previously and the weights a model's contribution by its performance.