Lab1_Classification

Lab 1_ Classification

Method 1: Tree based classification

step 1: collecting data

```
getwd()
```

```
## [1] "C:/Users/Madhu/Side Projects/Machine_Learning_Course"
```

```
setwd('C:/Users/Madhu/Side Projects/Machine_Learning_Course')

#Read data file

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 ...
```

Step 2: Exploring the data

```
#Using sumary to check the statistics of amunt column in the dataset

summary(credit$amount)

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

str(credit$default)

##  Factor w/ 2 levels "no","yes": 1 2 1 1 2 1 1 1 1 2 ...

#Default has two levels - yes or no
#Using table to see how many were defaulted and how many were not

table(credit$default)

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

Steps to develop tree based classification

```
#Before creating the testing and training data, randomizing the observations

set.seed(12345)

credit_rand <- credit[order(runif(1000)),]

#checking the summary of the original data with the randomized one to notice
any substantial changes

summary(credit$amount)

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

summary(credit_rand$amount)

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

Splitting data to training and testing set

```
# Choosing 90% for training set and remaining 10% as the testing set

credit_train <- credit_rand[1:900,]
```

```
credit_test <- credit_rand[901:1000,]

#Looking at percentage split of the testing and training to see if
randomization went well

prop.table(table(credit_train$default))

##
##        no       yes
## 0.7022222 0.2977778

prop.table(table(credit_train$default))

##
##        no       yes
## 0.7022222 0.2977778
```

Training a model on the data

```
#install.packages('C50')
library(C50)

## Warning: package 'C50' was built under R version 3.5.2

# Buiding ecision tree. Since we are predicting defaulted or not, we must
specify the 17th column to represent it as the class or response variable

credit_model <- C5.0(x = credit_train[-17], y = 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: 16
##
## Tree size: 67
##
## Non-standard options: attempt to group attributes
```

Examining the decision tree

```
summary(credit_model)

##
## Call:
## C5.0.default(x = credit_train[-17], y = credit_train$default)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Feb 17 20:45:06 2019
```

```
## ------------------------------
##
## Class specified by attribute `outcome'
##
## Read 900 cases (17 attributes) from undefined.data
##
## Decision tree:
##
## checking_balance = unknown: no (358/44)
## checking_balance in {< 0 DM,> 200 DM,1 - 200 DM}:
## :...credit_history in {perfect,very good}:
##     :...dependents > 1: yes (10/1)
##     :   dependents <= 1:
##     :   :...savings_balance = < 100 DM: yes (39/11)
##     :       savings_balance in {> 1000 DM,500 - 1000 DM,unknown}: no (8/1)
##     :       savings_balance = 100 - 500 DM:
##     :       :...checking_balance = < 0 DM: no (1)
##     :           checking_balance in {> 200 DM,1 - 200 DM}: yes (5/1)
##     credit_history in {critical,good,poor}:
##     :...months_loan_duration <= 11: no (87/14)
##         months_loan_duration > 11:
##         :...savings_balance = > 1000 DM: no (13)
##             savings_balance in {< 100 DM,100 - 500 DM,500 - 1000
DM,unknown}:
##             :...checking_balance = > 200 DM:
##                 :...dependents > 1: yes (3)
##                 :   dependents <= 1:
##                 :   :...credit_history in {good,poor}: no (23/3)
##                 :       credit_history = critical:
##                 :       :...amount <= 2337: yes (3)
##                 :           amount > 2337: no (6)
##                 checking_balance = 1 - 200 DM:
##                 :...savings_balance = unknown: no (34/6)
##                 :   savings_balance in {< 100 DM,100 - 500 DM,500 - 1000
DM}:
##                 :   :...months_loan_duration > 45: yes (11/1)
##                 :       months_loan_duration <= 45:
##                 :       :...other_credit = store:
##                 :           :...age <= 35: yes (4)
##                 :           :   age > 35: no (2)
##                 :           other_credit = bank:
##                 :           :...years_at_residence <= 1: no (3)
##                 :           :   years_at_residence > 1:
##                 :           :   :...existing_loans_count <= 1: yes (5)
##                 :           :       existing_loans_count > 1:
##                 :           :       :...percent_of_income <= 2: no (4/1)
##                 :           :           percent_of_income > 2: yes (3)
##                 :           other_credit = none:
##                 :           :...job = unemployed: no (1)
##                 :               job = management:
```

```
##                   :               :...amount <= 7511: no (10/3)
##                   :               :   amount > 7511: yes (7)
##                   :               job = unskilled: [S1]
##                   :               job = skilled:
##                   :               :...dependents <= 1: no (55/15)
##                   :                   dependents > 1:
##                   :                   :...age <= 34: no (3)
##                   :                       age > 34: yes (4)
##                   checking_balance = < 0 DM:
##                   :...job = management: no (26/6)
##                       job = unemployed: yes (4/1)
##                       job = unskilled:
##                       :...employment_duration in {4 - 7 years,
##                       :   :                       unemployed}: no (4)
##                       :   employment_duration = < 1 year:
##                       :   :...other_credit = bank: no (1)
##                       :   :   other_credit in {none,store}: yes (11/2)
##                       :   employment_duration = > 7 years:
##                       :   :...other_credit in {bank,none}: no (5/1)
##                       :   :   other_credit = store: yes (2)
##                       :   employment_duration = 1 - 4 years:
##                       :   :...age <= 39: no (14/3)
##                       :       age > 39:
##                       :       :...credit_history in {critical,good}: yes (3)
##                       :           credit_history = poor: no (1)
##                       job = skilled:
##                       :...credit_history = poor:
##                           :...savings_balance in {< 100 DM,100 - 500 DM,
##                           :   :                   500 - 1000 DM}: yes (8)
##                           :   savings_balance = unknown: no (1)
##                           credit_history = critical:
##                           :...other_credit = store: no (0)
##                           :   other_credit = bank: yes (4)
##                           :   other_credit = none:
##                           :   :...savings_balance in {100 - 500 DM,
##                           :       :                   unknown}: no (1)
##                           :       savings_balance = 500 - 1000 DM: yes (1)
##                           :       savings_balance = < 100 DM:
##                           :       :...months_loan_duration <= 13:
##                           :           :...percent_of_income <= 3: yes (3)
##                           :           :   percent_of_income > 3: no (3/1)
##                           :           months_loan_duration > 13:
##                           :           :...amount <= 5293: no (10/1)
##                           :               amount > 5293: yes (2)
##                           credit_history = good:
##                           :...existing_loans_count > 1: yes (5)
##                               existing_loans_count <= 1:
##                               :...other_credit = store: no (2)
##                                   other_credit = bank:
##                                   :...percent_of_income <= 2: yes (2)
```

```
##                                   :   percent_of_income > 2: no (6/1)
##                                   other_credit = none: [S2]
##
## SubTree [S1]
##
## employment_duration in {< 1 year,1 - 4 years}: yes (11/3)
## employment_duration in {> 7 years,4 - 7 years,unemployed}: no (8)
##
## SubTree [S2]
##
## savings_balance = 100 - 500 DM: yes (3)
## savings_balance = 500 - 1000 DM: no (1)
## savings_balance = unknown:
## :...phone = no: yes (9/1)
## :   phone = yes: no (3/1)
## savings_balance = < 100 DM:
## :...percent_of_income <= 1: no (4)
##     percent_of_income > 1:
##     :...phone = yes: yes (10/1)
##         phone = no:
##         :...purpose in {business,car0,education,renovations}: yes (3)
##             purpose = car:
##             :...percent_of_income <= 3: no (2)
##             :   percent_of_income > 3: yes (6/1)
##             purpose = furniture/appliances:
##             :...years_at_residence <= 1: no (4)
##                 years_at_residence > 1:
##                 :...housing = other: no (1)
##                     housing = rent: yes (2)
##                     housing = own:
##                     :...amount <= 1778: no (3)
##                         amount > 1778:
##                         :...years_at_residence <= 3: yes (6)
##                             years_at_residence > 3: no (3/1)
##
##
## Evaluation on training data (900 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      66  125(13.9%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     609    23    (a): class no
##     102   166    (b): class yes
##
```

```
##
##  Attribute usage:
##
##  100.00% checking_balance
##   60.22% credit_history
##   53.22% months_loan_duration
##   49.44% savings_balance
##   30.89% job
##   25.89% other_credit
##   17.78% dependents
##    9.67% existing_loans_count
##    7.22% percent_of_income
##    6.67% employment_duration
##    5.78% phone
##    5.56% amount
##    3.78% years_at_residence
##    3.44% age
##    3.33% purpose
##    1.67% housing
##
##
## Time: 0.0 secs
```

As we can see, 590 were classified as class a, 166 were classified as class b . There were 125 misclassified.

step 4: Evaluating Model performance

```
cred_pred <- predict(credit_model, credit_test)
# using gmodels ro create confusion matrix

#install.packages('gmodels')

library(gmodels)

## Warning: package 'gmodels' was built under R version 3.5.2

CrossTable(credit_test$default, cred_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 |        57 |        11 |        68 |
##                |     0.570 |     0.110 |           |
## ---------------|-----------|-----------|-----------|
##            yes |        16 |        16 |        32 |
##                |     0.160 |     0.160 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |        73 |        27 |       100 |
## ---------------|-----------|-----------|-----------|
##
##
```

As seen from the above confusion matrix, 11 were misclassified as a Type 2 error and 16 were type 1 that is it was yes but classifed as no.

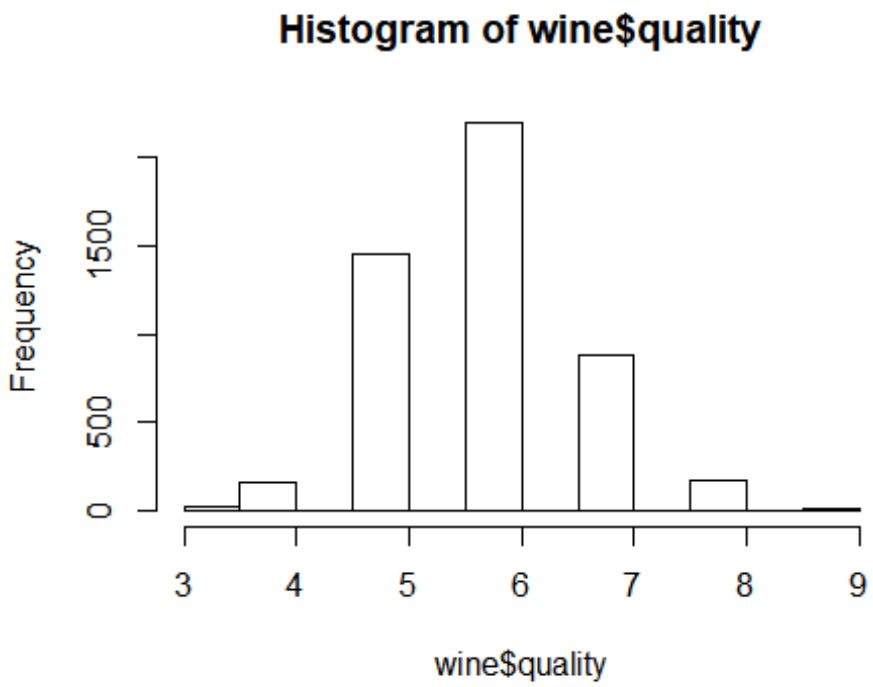
Method#2: Adding Regression to trees

Step 1: Collecting data

```
# Read data
wine <- read.csv("whitewines.csv")
str(wine)
```

```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  6.7 5.7 5.9 5.3 6.4 7 7.9 6.6 7 6.5 ...
##  $ volatile.acidity    : num  0.62 0.22 0.19 0.47 0.29 0.14 0.12 0.38 0.16
0.37 ...
##  $ citric.acid         : num  0.24 0.2 0.26 0.1 0.21 0.41 0.49 0.28 0.3
0.33 ...
##  $ residual.sugar      : num  1.1 16 7.4 1.3 9.65 0.9 5.2 2.8 2.6 3.9 ...
##  $ chlorides           : num  0.039 0.044 0.034 0.036 0.041 0.037 0.049
0.043 0.043 0.027 ...
##  $ free.sulfur.dioxide : num  6 41 33 11 36 22 33 17 34 40 ...
##  $ total.sulfur.dioxide: num  62 113 123 74 119 95 152 67 90 130 ...
##  $ density             : num  0.993 0.999 0.995 0.991 0.993 ...
##  $ pH                  : num  3.41 3.22 3.49 3.48 2.99 3.25 3.18 3.21 2.88
3.28 ...
##  $ sulphates           : num  0.32 0.46 0.42 0.54 0.34 0.43 0.47 0.47 0.47
0.39 ...
##  $ alcohol             : num  10.4 8.9 10.1 11.2 10.9 ...
##  $ quality             : int  5 6 6 4 6 6 6 6 6 7 ...
```

```
# Checkimg the distribution of quality variable to see if its normal
hist(wine$quality)
```



Yes, the distribution seems pretty normal and we can use regression on this class variable.

Next, we explore and prepare data in step 2.

```
# Creating training and testing data set
# 75% for tarining and 25% for testing

wine_train <- wine[1:3750, ]
wine_test <- wine[3751:4898, ]
```

Step 3, training model on data

```
# Installing rpart - recursive partitioning

#install.packages('rpart')
library(rpart)

## Warning: package 'rpart' was built under R version 3.5.2

m.rpart <- rpart(quality ~ ., data=wine_train)
m.rpart

## n= 3750
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 3750 2945.53200 5.870933
```

```
##    2) alcohol< 10.85 2372 1418.86100 5.604975
##      4) volatile.acidity>=0.2275 1611  821.30730 5.432030
##        8) volatile.acidity>=0.3025 688  278.97670 5.255814 *
##        9) volatile.acidity< 0.3025 923  505.04230 5.563380 *
##      5) volatile.acidity< 0.2275 761  447.36400 5.971091 *
##    3) alcohol>=10.85 1378 1070.08200 6.328737
##      6) free.sulfur.dioxide< 10.5 84   95.55952 5.369048 *
##      7) free.sulfur.dioxide>=10.5 1294  892.13600 6.391036
##       14) alcohol< 11.76667 629  430.11130 6.173291
##         28) volatile.acidity>=0.465 11   10.72727 4.545455 *
##         29) volatile.acidity< 0.465 618  389.71680 6.202265 *
##       15) alcohol>=11.76667 665  403.99400 6.596992 *
```

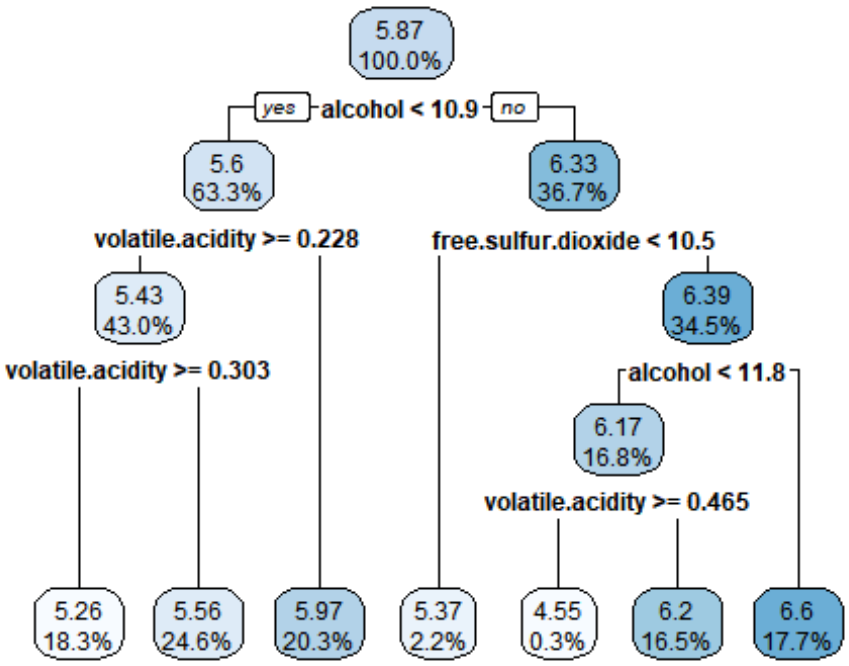
```
#install.packages('rpart.plot')

library(rpart.plot)
```

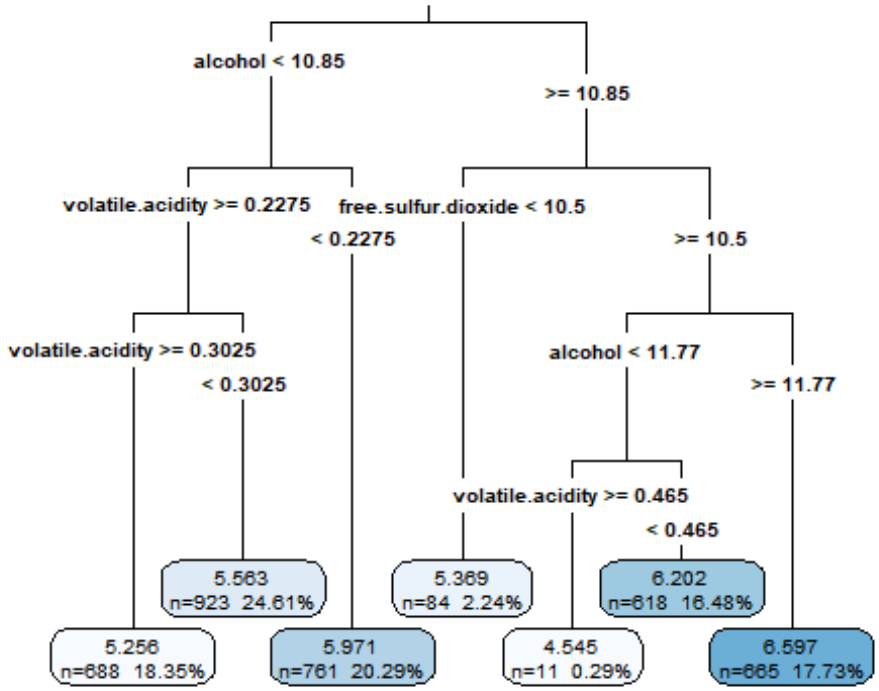
```
## Warning: package 'rpart.plot' was built under R version 3.5.2
```

```
#Visualizing the tree

rpart.plot(m.rpart, digits=3)
```




```
rpart.plot(m.rpart, digits=4, fallen.leaves = TRUE, type = 3, extra = 101)
```



The last and final step 4 of evaluating model performance

```
p.rpart <- predict(m.rpart, wine_test)
summary(p.rpart)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.545   5.563   5.971   5.893   6.202   6.597

summary(wine_test$quality)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.901   6.000   9.000

cor(p.rpart, wine_test$quality)

## [1] 0.5369525
```

A 54% correlation is seen.