2018R2 Data Mining (STAT5104) Assignment 1 Q2

Yiu Chung WONG 1155017920

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
library(caret)
library(rpart)
library(rpart.plot)
```

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

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.5.2
```

```
set.seed(12345)
```

###a)

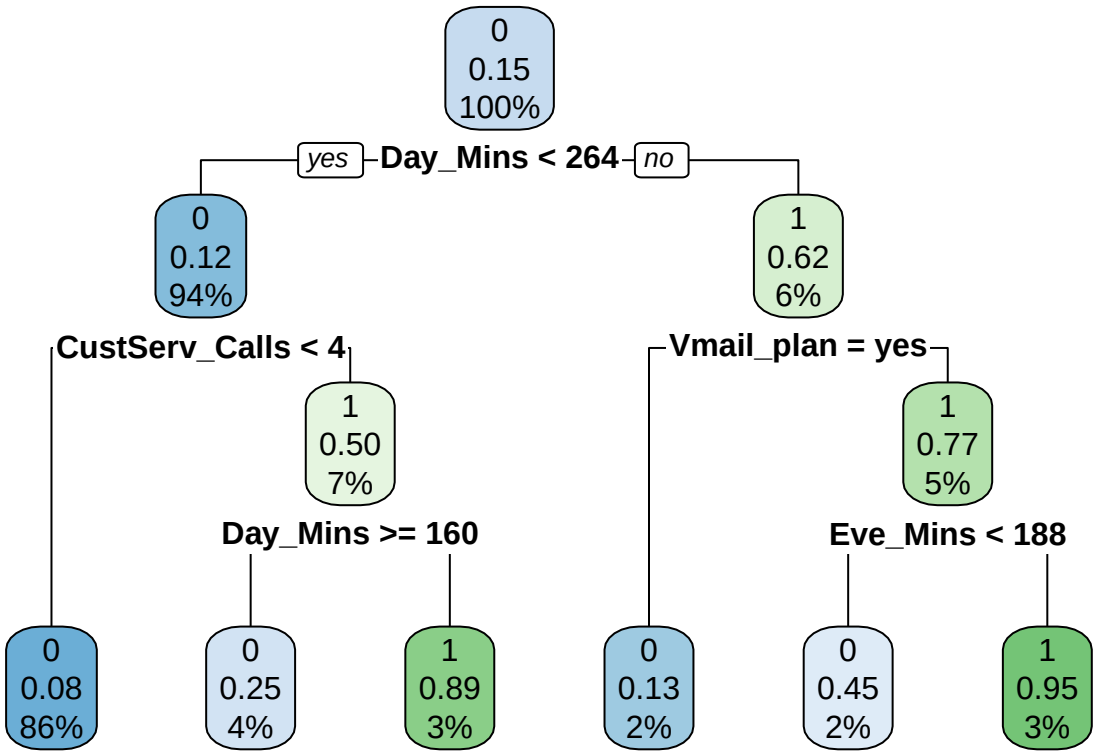
```
dc <- read.csv("tele.csv", header = TRUE, sep = ",")             #read data
inTrain <- createDataPartition(dc$Change, p = .9) %>% unlist(.) #create index for train / test partitio
d0 <- dc[inTrain,]                                               #select observations for training
d1 <- dc[-inTrain,]                                              #select observations for testing
```

###b)

```
control <- rpart.control(maxdepth = 3)
ctree <- rpart(data = d0, formula = Change ~ ., control = control, method = 'class')
```

###c)

```
rpart.plot(ctree)              #print plot
```




```
print(ctree)
```

```
## n= 2960
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 2960 439 0 (0.85168919 0.14831081)
##    2) Day_Mins< 264.45 2774 324 0 (0.88320115 0.11679885)
##      4) CustServ_Calls< 3.5 2556 214 0 (0.91627543 0.08372457) *
##      5) CustServ_Calls>=3.5 218 108 1 (0.49541284 0.50458716)
##       10) Day_Mins>=160.2 130  32 0 (0.75384615 0.24615385) *
##       11) Day_Mins< 160.2 88  10 1 (0.11363636 0.88636364) *
##    3) Day_Mins>=264.45 186  71 1 (0.38172043 0.61827957)
##      6) Vmail_plan=yes 45   6 0 (0.86666667 0.13333333) *
##      7) Vmail_plan=no 141  32 1 (0.22695035 0.77304965)
##       14) Eve_Mins< 187.75 49  22 0 (0.55102041 0.44897959) *
##       15) Eve_Mins>=187.75 92   5 1 (0.05434783 0.94565217) *
```

```
asRules(ctree, compact=FALSE) #print rules
```

```
##
##  Rule number: 15 [Change=1 cover=92 (3%) prob=0.95]
##    Day_Mins>=264.4
```

```
##    Vmail_plan=no
##    Eve_Mins>=187.8
##
##  Rule number: 11 [Change=1 cover=88 (3%) prob=0.89]
##    Day_Mins< 264.4
##    CustServ_Calls>=3.5
##    Day_Mins< 160.2
##
##  Rule number: 14 [Change=0 cover=49 (2%) prob=0.45]
##    Day_Mins>=264.4
##    Vmail_plan=no
##    Eve_Mins< 187.8
##
##  Rule number: 10 [Change=0 cover=130 (4%) prob=0.25]
##    Day_Mins< 264.4
##    CustServ_Calls>=3.5
##    Day_Mins>=160.2
##
##  Rule number: 6 [Change=0 cover=45 (2%) prob=0.13]
##    Day_Mins>=264.4
##    Vmail_plan=yes
##
##  Rule number: 4 [Change=0 cover=2556 (86%) prob=0.08]
##    Day_Mins< 264.4
##    CustServ_Calls< 3.5
```

```
oneSum <- sum(d0$Change)       #calculate total number of observaation where Change == 1
                               #in the training set
zeroSum <- nrow(d0) - oneSum   #calculate total number of observaation where Change == 0
                               #in the training set
```

Rule Number 4: Day_Mins< 264.45 and CustServ_Calls< 3.5 then Change = 0 Support = 2563 / 3288 = 0.7795012 Confidence = 1 - (198 / 2563) = 0.9227468 Capture = (2563 - 198) / 2542 = 0.9381198

Rule Number 10: Day_Mins< 264.45 and CustServ_Calls>= 3.5 and Day_Mins>=160.2 then Change = 0 Support = 130 / 3288 = 0.0395377 Confidence = 1 - (32 / 130) = 0.7538462 Capture = (130 - 32) / 2542 = 0.0388735

Rule Number 11: Day_Mins< 264.45 and CustServ_Calls>= 3.5 and Day_Mins< 160.2 then Change = 1 Support = 88 / 3288 = 0.026764 Confidence = 1 - (10 / 88) = 0.8863636 Capture = (88 - 10) / 418 = 0.2232346

Rule Number 6: Day_Mins>=264.45 and Vmail_plan=yes then Change = 0 Support = 44 / 3288 = 0.013382 Confidence = 1 - (6 / 44) = 0.8636364 Capture = (44 - 6) / 2542 = 0.0150734

Rule Number 14: Day_Mins>=264.45 and Vmail_plan=yes and then Eve_Mins<187.75 Change = 0 Support = 47 / 3288 = 0.0142944 Confidence = 1 - (21 / 47) = 0.5531915 Capture = (47 - 21) / 2542 = 0.0103134

Rule Number 15: Day_Mins>=264.45 and Vmail_plan=yes and then Eve_Mins>=187.75 Change = 1 Support = 88 / 3288 = 0.026764 Confidence = 1 - (5 / 88) = 0.9431818 Capture = (88 - 5) / 2542 = 0.1890661

###d)

```
test <- predict(ctree)                          #predict using training data
cl_test <- max.col(test) - 1                    #rename columns from  1, 2 to 0, 1
test_table <- table(cl_test, d0$Change)         #put data into classification table
test_table
```

```
##
## cl_test    0    1
##       0 2506  274
##       1   15  165
```

```
validation <- predict(ctree, newdata = d1)           #predict using testing data
cl_validation <- max.col(validation) - 1             #rename columns from  1, 2 to 0, 1
validation_table <- table(cl_validation, d1$Change) #put data into classification table
validation_table
```

```
##
## cl_validation   0   1
##             0 291  20
##             1   1  16
```

d0 error rate: (15 + 274) / 2960 = 0.0976351

d1 error rate: (1 + 20) / 328 = 0.0640244