akhilasaineni_Lab1

Akhila Saineni

10/18/2020

R Markdown

1 Tree- Based Classification

```
credit <- read.csv("/Users/akhilasaineni/Downloads/HU/2020Fall/ANLY_530_MachineLearning1/Lab1/credit.csv
str(credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ Creditability                    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Account.Balance                  : int  1 1 2 1 1 1 1 1 4 2 ...
##  $ Duration.of.Credit..month.       : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ Payment.Status.of.Previous.Credit: int  4 4 2 4 4 4 4 4 4 2 ...
##  $ Purpose                          : int  2 0 9 0 0 0 0 0 3 3 ...
##  $ Credit.Amount                    : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ Value.Savings.Stocks             : int  1 1 2 1 1 1 1 1 1 3 ...
##  $ Length.of.current.employment     : int  2 3 4 3 3 2 4 2 1 1 ...
##  $ Instalment.per.cent              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ Sex...Marital.Status             : int  2 3 2 3 3 3 3 3 2 2 ...
##  $ Guarantors                       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Duration.in.Current.address      : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ Most.valuable.available.asset    : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ Age..years.                      : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ Concurrent.Credits               : int  3 3 3 3 1 3 3 3 3 3 ...
##  $ Type.of.apartment                : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ No.of.Credits.at.this.Bank       : int  1 2 1 2 2 2 2 1 2 1 ...
##  $ Occupation                       : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ No.of.dependents                 : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ Telephone                        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Foreign.Worker                   : int  1 1 1 2 2 2 2 2 1 1 ...
```

```
summary(credit$Credit.Amount)
```

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

```
table(credit$Creditability)
```

```
##
##   0   1
## 300 700
```

```r
#Creating random
set.seed(12345)
credit_rand <- credit[order(runif(1000)), ]
summary(credit$ Credit.Amount)
```

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

```r
credit_train <- credit_rand[1:900, ]
credit_test <- credit_rand[901:1000, ]

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

```
##
##         0         1
## 0.3088889 0.6911111
```

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

```
##
##    0    1
## 0.22 0.78
```

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

credit_model <- C5.0(x = credit_train[-1], y = as.factor(credit_train$Creditability))
summary(credit_model)
```

```
##
## Call:
## C5.0.default(x = credit_train[-1], y = as.factor(credit_train$Creditability))
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Oct 18 19:40:27 2020
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 900 cases (21 attributes) from undefined.data
##
## Decision tree:
##
## Account.Balance > 2:
## :...Concurrent.Credits > 2:
## :   :...Age..years. > 33: 1 (179/11)
## :   :   Age..years. <= 33:
## :   :   :...Credit.Amount > 6681:
## :   :       :...Length.of.current.employment <= 2: 0 (4)
## :   :       :   Length.of.current.employment > 2:
## :   :       :   :...Payment.Status.of.Previous.Credit <= 3: 1 (4)
```

```
## :   :       :       Payment.Status.of.Previous.Credit > 3: 0 (3/1)
## :   :       Credit.Amount <= 6681:
## :   :       :...Occupation > 2:
## :   :           :...Occupation <= 3: 1 (120/12)
## :   :           :   Occupation > 3:
## :   :           :   :...Duration.of.Credit..month. <= 33: 1 (9)
## :   :           :       Duration.of.Credit..month. > 33: 0 (3)
## :   :           Occupation <= 2:
## :   :           :...No.of.Credits.at.this.Bank > 1: 1 (6)
## :   :               No.of.Credits.at.this.Bank <= 1:
## :   :               :...Most.valuable.available.asset > 1: 0 (3)
## :   :                   Most.valuable.available.asset <= 1:
## :   :                   :...Credit.Amount <= 1987: 1 (8/1)
## :   :                       Credit.Amount > 1987: 0 (2)
## :   Concurrent.Credits <= 2:
## :   :...Guarantors > 1: 1 (4)
## :       Guarantors <= 1:
## :       :...Purpose <= 0:
## :           :...Most.valuable.available.asset <= 2: 0 (5)
## :           :   Most.valuable.available.asset > 2:
## :           :   :...No.of.dependents <= 1: 1 (7/1)
## :           :       No.of.dependents > 1: 0 (2)
## :           Purpose > 0:
## :           :...Purpose <= 4: 1 (35/2)
## :               Purpose > 4:
## :               :...Length.of.current.employment <= 2: 0 (4)
## :                   Length.of.current.employment > 2:
## :                   :...No.of.dependents > 1: 0 (3/1)
## :                       No.of.dependents <= 1:
## :                       :...Length.of.current.employment > 3: 1 (4)
## :                           Length.of.current.employment <= 3:
## :                           :...Instalment.per.cent <= 2: 1 (2)
## :                               Instalment.per.cent > 2: 0 (2)
## Account.Balance <= 2:
## :...Payment.Status.of.Previous.Credit <= 1:
##     :...Value.Savings.Stocks <= 2: 0 (49/10)
##     :   Value.Savings.Stocks > 2:
##     :   :...Credit.Amount <= 2064: 0 (3)
##     :       Credit.Amount > 2064: 1 (9/1)
##     Payment.Status.of.Previous.Credit > 1:
##     :...Credit.Amount > 7980:
##         :...Value.Savings.Stocks > 4:
##         :   :...Payment.Status.of.Previous.Credit <= 2: 0 (4/1)
##         :   :   Payment.Status.of.Previous.Credit > 2: 1 (3)
##         :   Value.Savings.Stocks <= 4:
##         :   :...Account.Balance > 1: 0 (15)
##         :       Account.Balance <= 1:
##         :       :...Concurrent.Credits <= 2: 0 (2)
##         :           Concurrent.Credits > 2:
##         :           :...Credit.Amount <= 10297: 0 (6)
##         :               Credit.Amount > 10297: 1 (3)
##         Credit.Amount <= 7980:
##         :...Duration.of.Credit..month. <= 11:
##             :...Occupation > 3:
```

```
##             :   :...Concurrent.Credits <= 2: 1 (3)
##             :   :   Concurrent.Credits > 2:
##             :   :   :...Payment.Status.of.Previous.Credit <= 2: 1 (4/1)
##             :   :       Payment.Status.of.Previous.Credit > 2: 0 (3)
##             :   Occupation <= 3:
##             :   :...Age..years. > 32: 1 (34)
##             :       Age..years. <= 32:
##             :       :...Most.valuable.available.asset <= 1: 1 (13/1)
##             :           Most.valuable.available.asset > 1:
##             :           :...Instalment.per.cent <= 3: 1 (6/1)
##             :               Instalment.per.cent > 3: 0 (6/1)
##             Duration.of.Credit..month. > 11:
##             :...Duration.of.Credit..month. > 36:
##                 :...Length.of.current.employment <= 1: 1 (3)
##                 :   Length.of.current.employment > 1:
##                 :   :...No.of.dependents > 1: 1 (5/1)
##                 :       No.of.dependents <= 1:
##                 :       :...Duration.in.Current.address <= 1: 1 (4/1)
##                 :           Duration.in.Current.address > 1: 0 (23)
##                 Duration.of.Credit..month. <= 36:
##                 :...Guarantors > 2:
##                     :...Foreign.Worker <= 1: 1 (23/1)
##                     :   Foreign.Worker > 1: 0 (2)
##                     Guarantors <= 2:
##                     :...Credit.Amount <= 1381:
##                         :...Telephone > 1:
##                         :   :...Sex...Marital.Status > 3: 0 (2)
##                         :   :   Sex...Marital.Status <= 3:
##                         :   :   :...Duration.of.Credit..month. <= 16: 1 (7)
##                         :   :       Duration.of.Credit..month. > 16: 0 (3/1)
##                         :   Telephone <= 1:
##                         :   :...Concurrent.Credits <= 2: 0 (9)
##                         :       Concurrent.Credits > 2:
##                         :       :...Account.Balance <= 1: 0 (29/6)
##                         :           Account.Balance > 1: [S1]
##                         Credit.Amount > 1381:
##                         :...Guarantors > 1:
##                             :...Foreign.Worker > 1: 1 (2)
##                             :   Foreign.Worker <= 1:
##                             :   :...Instalment.per.cent > 2: 0 (5)
##                             :       Instalment.per.cent <= 2: [S2]
##                             Guarantors <= 1:
##                             :...Payment.Status.of.Previous.Credit > 3:
##                                 :...Age..years. > 33: 1 (22)
##                                 :   Age..years. <= 33:
##                                 :   :...Purpose > 3: 1 (7)
##                                 :       Purpose <= 3: [S3]
##                                 Payment.Status.of.Previous.Credit <= 3:
##                                 :...Instalment.per.cent <= 2:
##                                     :...No.of.dependents > 1:
##                                     :   :...Purpose <= 0: 1 (2)
##                                     :   :   Purpose > 0: 0 (3)
##                                     :   No.of.dependents <= 1: [S4]
##                                     Instalment.per.cent > 2:
```

```
##                                     :...Concurrent.Credits <= 1: 1 (8/1)
##                                         Concurrent.Credits > 1:
##                                         :...Sex...Marital.Status <= 1: 0 (6/1)
##                                             Sex...Marital.Status > 1:
##                                             :...Account.Balance > 1: [S5]
##                                                 Account.Balance <= 1: [S6]
##
## SubTree [S1]
##
## Duration.in.Current.address > 3: 1 (8/1)
## Duration.in.Current.address <= 3:
## :...Purpose > 2: 0 (5)
##     Purpose <= 2:
##     :...Type.of.apartment <= 1: 0 (2)
##         Type.of.apartment > 1: 1 (5/1)
##
## SubTree [S2]
##
## Duration.in.Current.address <= 2: 1 (2)
## Duration.in.Current.address > 2: 0 (4/1)
##
## SubTree [S3]
##
## Duration.of.Credit..month. <= 16: 1 (4)
## Duration.of.Credit..month. > 16:
## :...Length.of.current.employment <= 3: 0 (8)
##     Length.of.current.employment > 3: 1 (6/1)
##
## SubTree [S4]
##
## Duration.in.Current.address > 1: 1 (41/6)
## Duration.in.Current.address <= 1:
## :...Value.Savings.Stocks > 3: 0 (2)
##     Value.Savings.Stocks <= 3:
##     :...Length.of.current.employment > 2: 1 (4)
##         Length.of.current.employment <= 2:
##         :...Instalment.per.cent <= 1: 0 (3)
##             Instalment.per.cent > 1: 1 (3/1)
##
## SubTree [S5]
##
## Sex...Marital.Status > 3: 0 (2)
## Sex...Marital.Status <= 3:
## :...Length.of.current.employment > 3: 1 (10)
##     Length.of.current.employment <= 3:
##     :...Duration.in.Current.address <= 1: 1 (5)
##         Duration.in.Current.address > 1:
##         :...Length.of.current.employment <= 2: 0 (4)
##             Length.of.current.employment > 2:
##             :...Value.Savings.Stocks <= 1: 0 (3)
##                 Value.Savings.Stocks > 1: 1 (5)
##
## SubTree [S6]
##
```

```
## Payment.Status.of.Previous.Credit > 2: 0 (3)
## Payment.Status.of.Previous.Credit <= 2:
## :...Purpose <= 0: 0 (7/1)
##     Purpose > 0:
##     :...Most.valuable.available.asset <= 1: 0 (5/1)
##         Most.valuable.available.asset > 1:
##         :...Sex...Marital.Status <= 2: 1 (6)
##             Sex...Marital.Status > 2:
##             :...Length.of.current.employment > 4: 0 (5)
##                 Length.of.current.employment <= 4:
##                 :...Telephone > 1: 1 (3)
##                     Telephone <= 1:
##                     :...Length.of.current.employment <= 2: 0 (2)
##                         Length.of.current.employment > 2:
##                         :...Age..years. <= 28: 1 (4)
##                             Age..years. > 28: 0 (2)
##
##
## Evaluation on training data (900 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      85   70( 7.8%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     233    45    (a): class 0
##      25   597    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% Account.Balance
##   67.11% Credit.Amount
##   63.11% Concurrent.Credits
##   55.33% Payment.Status.of.Previous.Credit
##   50.33% Age..years.
##   45.44% Duration.of.Credit..month.
##   40.11% Guarantors
##   24.44% Occupation
##   18.33% Instalment.per.cent
##   15.56% Purpose
##   14.22% Length.of.current.employment
##   13.67% Duration.in.Current.address
##   12.67% Value.Savings.Stocks
##   12.22% No.of.dependents
##    9.33% Sex...Marital.Status
##    9.00% Telephone
##    8.78% Most.valuable.available.asset
##    4.22% Foreign.Worker
##    2.11% No.of.Credits.at.this.Bank
```

```
##    0.78% Type.of.apartment
##
##
## Time: 0.0 secs
```

```r
#install.packages("gmodels")
library(gmodels)
cred_pred <- predict(credit_model, credit_test)
CrossTable(credit_test$Creditability, cred_pred, prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c( 'Actual Creditability', 'Predicted Creditability'))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                      | Predicted Creditability
## Actual Creditability |         0 |         1 | Row Total |
## ---------------------|-----------|-----------|-----------|
##                    0 |         8 |        14 |        22 |
##                      |     0.080 |     0.140 |           |
## ---------------------|-----------|-----------|-----------|
##                    1 |        17 |        61 |        78 |
##                      |     0.170 |     0.610 |           |
## ---------------------|-----------|-----------|-----------|
##         Column Total |        25 |        75 |       100 |
## ---------------------|-----------|-----------|-----------|
##
##
```

Q1 If you see an accuracy of 100%, what does it mean? Does this mean that we design a perfect model? This is some thing that needs more discussion. Write a few sentences about accuracy of 100%.

When the accuracy of a model is 100% then it means that the model is able to predict accurately each and every single observation. This means that there is no Type 1 error or Type 2 error. On the other side, accuracy of 100% doesn't mean that the model is perfect because the model may have been overfitted or overtrained.

2 Random Forest

```r
#install.packages("randomForest")
library("randomForest")
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.
```

```
credit_train$Creditability <- as.factor(credit_train$Creditability)
random_model <- randomForest(Creditability ~ . , data= credit_train)
summary(random_model)
```

```
##                 Length Class  Mode
## call               3   -none- call
## type               1   -none- character
## predicted        900   factor numeric
## err.rate        1500   -none- numeric
## confusion          6   -none- numeric
## votes           1800   matrix numeric
## oob.times        900   -none- numeric
## classes            2   -none- character
## importance        20   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y                900   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
cred_pred <- predict(random_model, credit_test)
(p <- table(cred_pred, credit_test$Creditability))
```

```
##
## cred_pred  0  1
##         0 11 10
##         1 11 68
```

```
(Accuracy <- sum(diag(p))/sum(p)*100)
```

```
## [1] 79
```

```
importance(random_model)
```

```
##                                    MeanDecreaseGini
## Account.Balance                           42.599355
## Duration.of.Credit..month.                37.502785
## Payment.Status.of.Previous.Credit         22.563009
## Purpose                                   23.774048
## Credit.Amount                             52.397155
## Value.Savings.Stocks                      19.388385
## Length.of.current.employment              20.221289
```

```
## Instalment.per.cent                          16.394636
## Sex...Marital.Status                         13.424449
## Guarantors                                    7.475422
## Duration.in.Current.address                  15.563685
## Most.valuable.available.asset                17.326842
## Age..years.                                  37.377916
## Concurrent.Credits                            8.480725
## Type.of.apartment                             9.595344
## No.of.Credits.at.this.Bank                    8.424006
## Occupation                                   12.669816
## No.of.dependents                              5.774473
## Telephone                                     7.505291
## Foreign.Worker                                1.746964
```

Q2 What are the three most important features in this model.

The following are the most important features based on the Gini Score Account.Balance Duration.of.credit..month. Payment.status.of.previous.credit

```r
set.seed(23458)
random_model_seed_change <- randomForest(Creditability ~ . , data=credit_train)

cred_pred_seed_change <- predict(random_model_seed_change, credit_test)
p_seed_change <- table(cred_pred_seed_change, credit_test$Creditability)
(Accuracy_seed_change <- sum(diag(p_seed_change))/sum(p_seed_change)*100)
```

```
## [1] 80
```

```r
p_seed_change
```

```
##
## cred_pred_seed_change  0  1
##                     0 12 10
##                     1 10 68
```

The accuracy of the model with seed change remained close to the one with the previous seed 80% & 82% respectively.

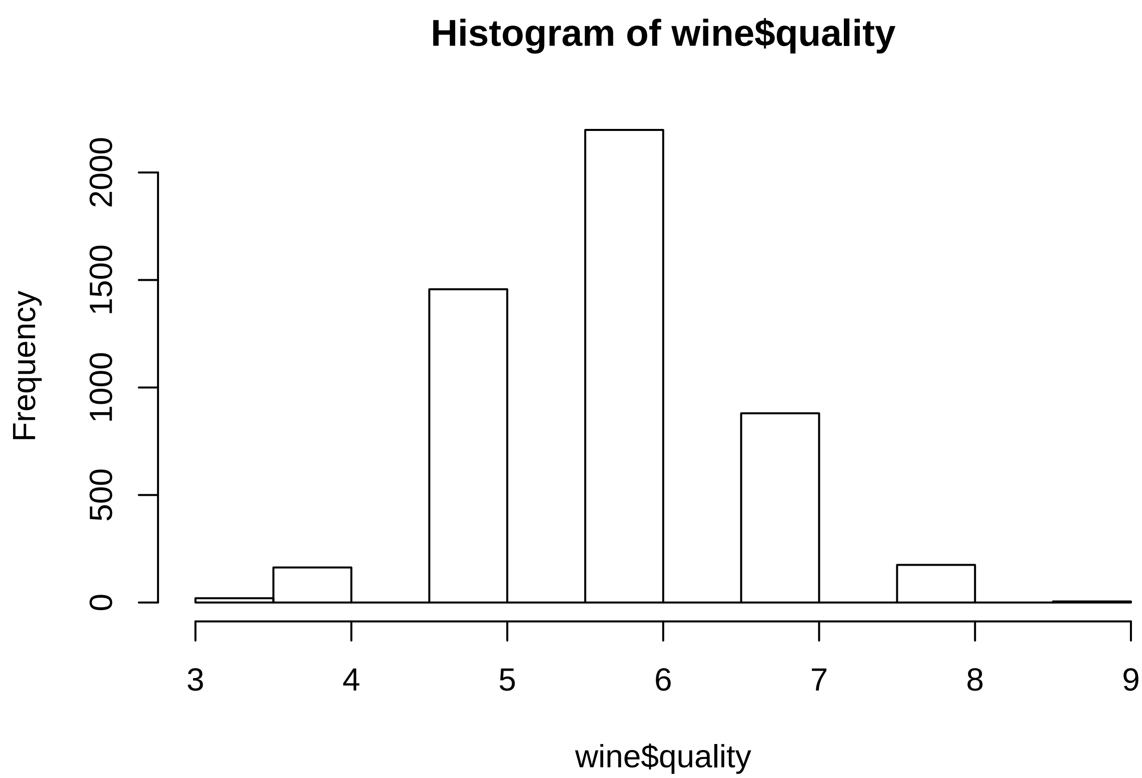
3 Adding Regression to Trees

```r
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 ...
```

```
hist(wine$quality)
```

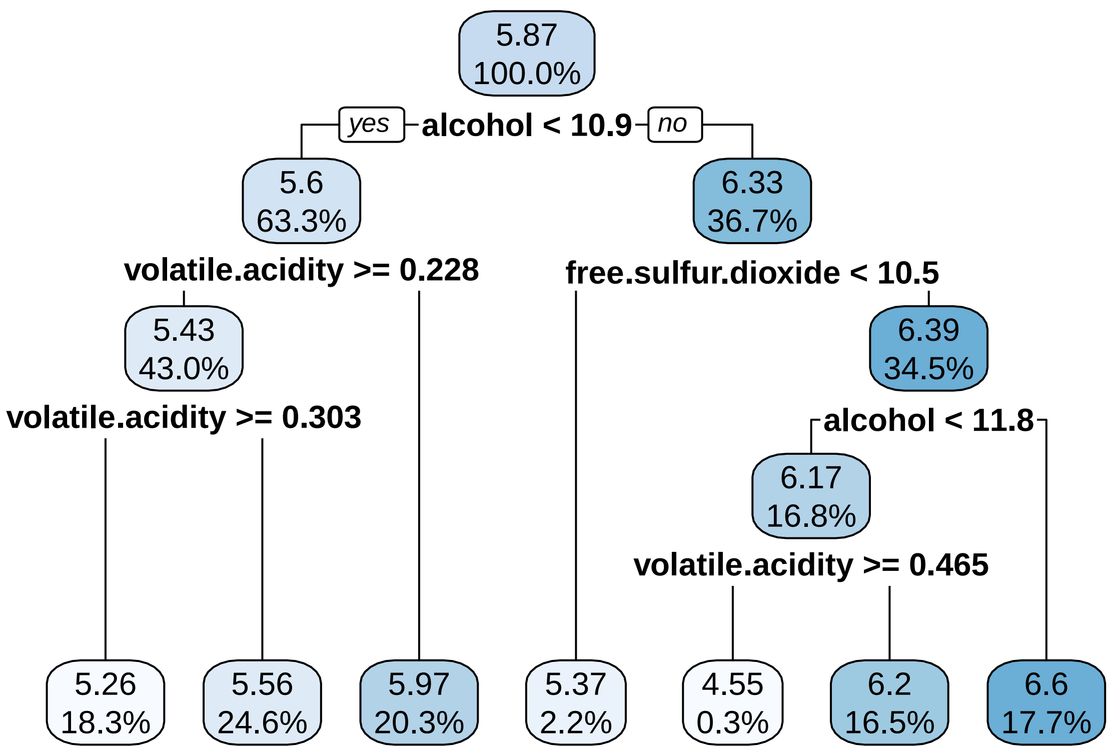



```
wine_train <- wine[1:3750, ]
wine_test <- wine[3751:4898, ]

#install.packages("rpart.plot")
library(rpart)

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

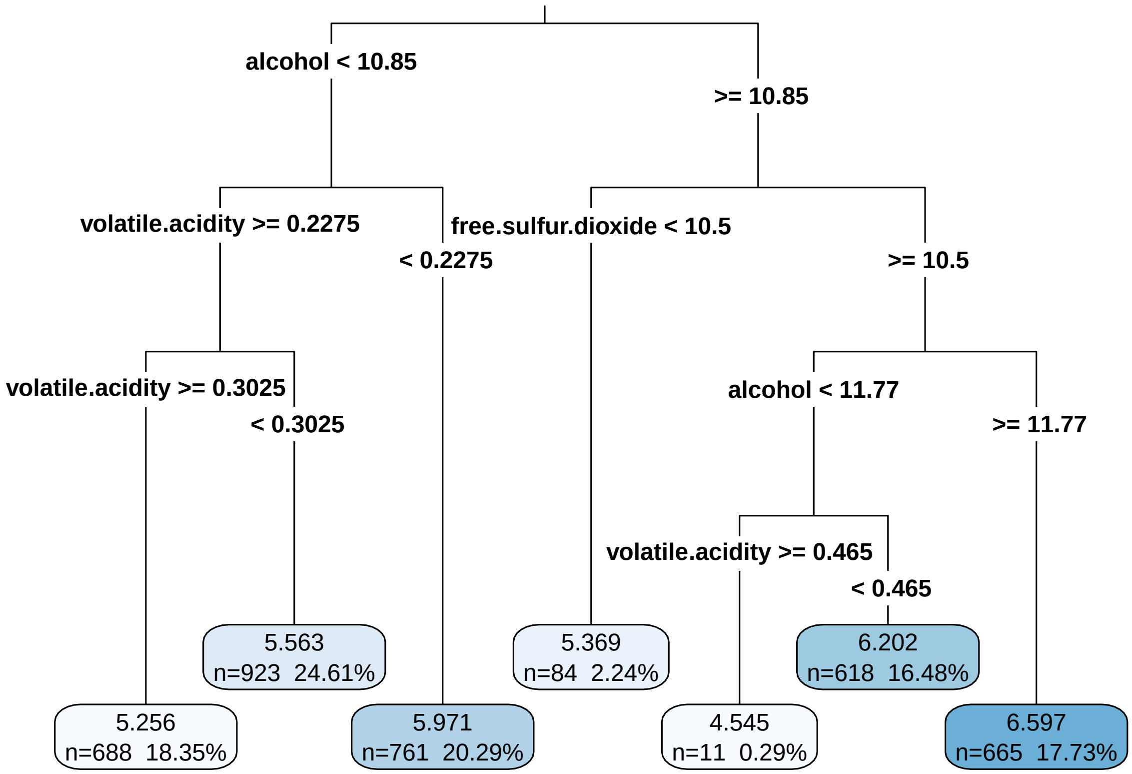
```
## n= 3750
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 3750 2945.53200 5.870933
##    2) alcohol< 10.85 2372 1418.86100 5.604975
```

```
##       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 *
```

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




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




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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.545   5.563   5.971   5.871   6.202   6.597
```

```r
summary(wine_test$quality)
```

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

Q3 What is your interpretation about this amount of RMSE?

The absolute measure of the fit is called the Root Mean Square Error. If the RMSE score is low that means that the predictions are close to the actual data whereas if the RMSE score is high, it means that the model is not predicting as expected.

4 News Popularity

```r
news_p<-read.csv("OnlineNewsPopularity_for_R.csv")
head(news_p)
```

```
##                                                              url timedelta
## 1   http://mashable.com/2013/01/07/amazon-instant-video-browser/       731
## 2    http://mashable.com/2013/01/07/ap-samsung-sponsored-tweets/       731
## 3 http://mashable.com/2013/01/07/apple-40-billion-app-downloads/       731
## 4       http://mashable.com/2013/01/07/astronaut-notre-dame-bcs/       731
## 5               http://mashable.com/2013/01/07/att-u-verse-apps/       731
## 6               http://mashable.com/2013/01/07/beewi-smart-toys/       731
##   n_tokens_title n_tokens_content n_unique_tokens n_non_stop_words
## 1             12              219       0.6635945                1
## 2              9              255       0.6047431                1
## 3              9              211       0.5751295                1
## 4              9              531       0.5037879                1
## 5             13             1072       0.4156456                1
## 6             10              370       0.5598886                1
##   n_non_stop_unique_tokens num_hrefs num_self_hrefs num_imgs num_videos
## 1                0.8153846         4              2        1          0
## 2                0.7919463         3              1        1          0
## 3                0.6638655         3              1        1          0
## 4                0.6656347         9              0        1          0
## 5                0.5408895        19             19       20          0
## 6                0.6981982         2              2        0          0
##   average_token_length num_keywords data_channel_is_lifestyle
## 1             4.680365            5                         0
## 2             4.913725            4                         0
## 3             4.393365            6                         0
## 4             4.404896            7                         0
## 5             4.682836            7                         0
## 6             4.359459            9                         0
##   data_channel_is_entertainment data_channel_is_bus data_channel_is_socmed
## 1                             1                   0                      0
## 2                             0                   1                      0
## 3                             0                   1                      0
## 4                             1                   0                      0
## 5                             0                   0                      0
## 6                             0                   0                      0
##   data_channel_is_tech data_channel_is_world kw_min_min kw_max_min kw_avg_min
## 1                    0                     0          0          0          0
## 2                    0                     0          0          0          0
## 3                    0                     0          0          0          0
## 4                    0                     0          0          0          0
## 5                    1                     0          0          0          0
## 6                    1                     0          0          0          0
##   kw_min_max kw_max_max kw_avg_max kw_min_avg kw_max_avg kw_avg_avg
## 1          0          0          0          0          0          0
## 2          0          0          0          0          0          0
## 3          0          0          0          0          0          0
## 4          0          0          0          0          0          0
## 5          0          0          0          0          0          0
## 6          0          0          0          0          0          0
##   self_reference_min_shares self_reference_max_shares
## 1                       496                       496
## 2                         0                         0
## 3                       918                       918
## 4                         0                         0
```

```
## 5                     545                   16000
## 6                    8500                    8500
##   self_reference_avg_sharess weekday_is_monday weekday_is_tuesday
## 1                    496.000                 1                  0
## 2                      0.000                 1                  0
## 3                    918.000                 1                  0
## 4                      0.000                 1                  0
## 5                   3151.158                 1                  0
## 6                   8500.000                 1                  0
##   weekday_is_wednesday weekday_is_thursday weekday_is_friday
## 1                    0                   0                 0
## 2                    0                   0                 0
## 3                    0                   0                 0
## 4                    0                   0                 0
## 5                    0                   0                 0
## 6                    0                   0                 0
##   weekday_is_saturday weekday_is_sunday is_weekend     LDA_00     LDA_01
## 1                   0                 0          0 0.50033120 0.37827893
## 2                   0                 0          0 0.79975569 0.05004668
## 3                   0                 0          0 0.21779229 0.03333446
## 4                   0                 0          0 0.02857322 0.41929964
## 5                   0                 0          0 0.02863281 0.02879355
## 6                   0                 0          0 0.02224528 0.30671758
##       LDA_02     LDA_03     LDA_04 global_subjectivity
## 1 0.04000468 0.04126265 0.04012254           0.5216171
## 2 0.05009625 0.05010067 0.05000071           0.3412458
## 3 0.03335142 0.03333354 0.68218829           0.7022222
## 4 0.49465083 0.02890472 0.02857160           0.4298497
## 5 0.02857518 0.02857168 0.88542678           0.5135021
## 6 0.02223128 0.02222429 0.62658158           0.4374086
##   global_sentiment_polarity global_rate_positive_words
## 1                0.09256198                 0.04566210
## 2                0.14894781                 0.04313725
## 3                0.32333333                 0.05687204
## 4                0.10070467                 0.04143126
## 5                0.28100348                 0.07462687
## 6                0.07118419                 0.02972973
##   global_rate_negative_words rate_positive_words rate_negative_words
## 1                0.013698630           0.7692308           0.2307692
## 2                0.015686275           0.7333333           0.2666667
## 3                0.009478673           0.8571429           0.1428571
## 4                0.020715631           0.6666667           0.3333333
## 5                0.012126866           0.8602151           0.1397849
## 6                0.027027027           0.5238095           0.4761905
##   avg_positive_polarity min_positive_polarity max_positive_polarity
## 1             0.3786364            0.10000000                   0.7
## 2             0.2869146            0.03333333                   0.7
## 3             0.4958333            0.10000000                   1.0
## 4             0.3859652            0.13636364                   0.8
## 5             0.4111274            0.03333333                   1.0
## 6             0.3506100            0.13636364                   0.6
##   avg_negative_polarity min_negative_polarity max_negative_polarity
## 1            -0.3500000                -0.600            -0.2000000
## 2            -0.1187500                -0.125            -0.1000000
```

```
## 3           -0.4666667                -0.800           -0.1333333
## 4           -0.3696970                -0.600           -0.1666667
## 5           -0.2201923                -0.500           -0.0500000
## 6           -0.1950000                -0.400           -0.1000000
##   title_subjectivity title_sentiment_polarity abs_title_subjectivity
## 1          0.5000000               -0.1875000             0.00000000
## 2          0.0000000                0.0000000             0.50000000
## 3          0.0000000                0.0000000             0.50000000
## 4          0.0000000                0.0000000             0.50000000
## 5          0.4545455                0.1363636             0.04545455
## 6          0.6428571                0.2142857             0.14285714
##   abs_title_sentiment_polarity shares
## 1                    0.1875000    593
## 2                    0.0000000    711
## 3                    0.0000000   1500
## 4                    0.0000000   1200
## 5                    0.1363636    505
## 6                    0.2142857    855
```

```r
str(news_p)
```

```
## 'data.frame':    39644 obs. of  61 variables:
##  $ url                          : Factor w/ 39644 levels "http://mashable.com/2013/01/07/amazon-insta
##  $ timedelta                    : num  731 731 731 731 731 731 731 731 731 731 ...
##  $ n_tokens_title               : num  12 9 9 9 13 10 8 12 11 10 ...
##  $ n_tokens_content             : num  219 255 211 531 1072 ...
##  $ n_unique_tokens              : num  0.664 0.605 0.575 0.504 0.416 ...
##  $ n_non_stop_words             : num  1 1 1 1 1 ...
##  $ n_non_stop_unique_tokens     : num  0.815 0.792 0.664 0.666 0.541 ...
##  $ num_hrefs                    : num  4 3 3 9 19 2 21 20 2 4 ...
##  $ num_self_hrefs               : num  2 1 1 0 19 2 20 20 0 1 ...
##  $ num_imgs                     : num  1 1 1 1 20 0 20 20 0 1 ...
##  $ num_videos                   : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ average_token_length         : num  4.68 4.91 4.39 4.4 4.68 ...
##  $ num_keywords                 : num  5 4 6 7 7 9 10 9 7 5 ...
##  $ data_channel_is_lifestyle    : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ data_channel_is_entertainment: num  1 0 0 1 0 0 0 0 0 0 ...
##  $ data_channel_is_bus          : num  0 1 1 0 0 0 0 0 0 0 ...
##  $ data_channel_is_socmed       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ data_channel_is_tech         : num  0 0 0 0 1 1 0 1 1 0 ...
##  $ data_channel_is_world        : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ kw_min_min                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_max_min                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_avg_min                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_min_max                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_max_max                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_avg_max                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_min_avg                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_max_avg                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ kw_avg_avg                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ self_reference_min_shares    : num  496 0 918 0 545 8500 545 545 0 0 ...
##  $ self_reference_max_shares    : num  496 0 918 0 16000 8500 16000 16000 0 0 ...
##  $ self_reference_avg_sharess   : num  496 0 918 0 3151 ...
##  $ weekday_is_monday            : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ weekday_is_tuesday           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday_is_wednesday         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday_is_thursday          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday_is_friday            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday_is_saturday          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday_is_sunday            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ is_weekend                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LDA_00                       : num  0.5003 0.7998 0.2178 0.0286 0.0286 ...
##  $ LDA_01                       : num  0.3783 0.05 0.0333 0.4193 0.0288 ...
##  $ LDA_02                       : num  0.04 0.0501 0.0334 0.4947 0.0286 ...
##  $ LDA_03                       : num  0.0413 0.0501 0.0333 0.0289 0.0286 ...
##  $ LDA_04                       : num  0.0401 0.05 0.6822 0.0286 0.8854 ...
##  $ global_subjectivity          : num  0.522 0.341 0.702 0.43 0.514 ...
##  $ global_sentiment_polarity    : num  0.0926 0.1489 0.3233 0.1007 0.281 ...
##  $ global_rate_positive_words   : num  0.0457 0.0431 0.0569 0.0414 0.0746 ...
##  $ global_rate_negative_words   : num  0.0137 0.01569 0.00948 0.02072 0.01213 ...
##  $ rate_positive_words          : num  0.769 0.733 0.857 0.667 0.86 ...
##  $ rate_negative_words          : num  0.231 0.267 0.143 0.333 0.14 ...
##  $ avg_positive_polarity        : num  0.379 0.287 0.496 0.386 0.411 ...
##  $ min_positive_polarity        : num  0.1 0.0333 0.1 0.1364 0.0333 ...
##  $ max_positive_polarity        : num  0.7 0.7 1 0.8 1 0.6 1 1 0.8 0.5 ...
##  $ avg_negative_polarity        : num  -0.35 -0.119 -0.467 -0.37 -0.22 ...
##  $ min_negative_polarity        : num  -0.6 -0.125 -0.8 -0.6 -0.5 -0.4 -0.5 -0.5 -0.125 -0.5 ...
##  $ max_negative_polarity        : num  -0.2 -0.1 -0.133 -0.167 -0.05 ...
##  $ title_subjectivity           : num  0.5 0 0 0 0.455 ...
##  $ title_sentiment_polarity     : num  -0.188 0 0 0 0.136 ...
##  $ abs_title_subjectivity       : num  0 0.5 0.5 0.5 0.0455 ...
##  $ abs_title_sentiment_polarity : num  0.188 0 0 0 0.136 ...
##  $ shares                       : int  593 711 1500 1200 505 855 556 891 3600 710 ...
```

```
colnames(news_p)
```

```
##  [1] "url"                           "timedelta"
##  [3] "n_tokens_title"                "n_tokens_content"
##  [5] "n_unique_tokens"               "n_non_stop_words"
##  [7] "n_non_stop_unique_tokens"      "num_hrefs"
##  [9] "num_self_hrefs"                "num_imgs"
## [11] "num_videos"                    "average_token_length"
## [13] "num_keywords"                  "data_channel_is_lifestyle"
## [15] "data_channel_is_entertainment" "data_channel_is_bus"
## [17] "data_channel_is_socmed"        "data_channel_is_tech"
## [19] "data_channel_is_world"         "kw_min_min"
## [21] "kw_max_min"                    "kw_avg_min"
## [23] "kw_min_max"                    "kw_max_max"
## [25] "kw_avg_max"                    "kw_min_avg"
## [27] "kw_max_avg"                    "kw_avg_avg"
## [29] "self_reference_min_shares"     "self_reference_max_shares"
## [31] "self_reference_avg_sharess"    "weekday_is_monday"
## [33] "weekday_is_tuesday"            "weekday_is_wednesday"
## [35] "weekday_is_thursday"           "weekday_is_friday"
## [37] "weekday_is_saturday"           "weekday_is_sunday"
## [39] "is_weekend"                    "LDA_00"
## [41] "LDA_01"                        "LDA_02"
## [43] "LDA_03"                        "LDA_04"
```

```
## [45] "global_subjectivity"          "global_sentiment_polarity"
## [47] "global_rate_positive_words"   "global_rate_negative_words"
## [49] "rate_positive_words"          "rate_negative_words"
## [51] "avg_positive_polarity"        "min_positive_polarity"
## [53] "max_positive_polarity"        "avg_negative_polarity"
## [55] "min_negative_polarity"        "max_negative_polarity"
## [57] "title_subjectivity"           "title_sentiment_polarity"
## [59] "abs_title_subjectivity"       "abs_title_sentiment_polarity"
## [61] "shares"
```

```
news_p <- news_p[,c("n_tokens_title", "n_tokens_content", "n_unique_tokens", "n_non_stop_words", "num_h

#We want to make this problem a classification one. One approach is to make any piece of article more t

#We will be using shares instead of likes
for(i in 1:39644) {
  news_p$fav[i]<- if( news_p$shares[i]>=1400) {"YES"} else {"NO"}
  }

head(news_p)
```

```
##   n_tokens_title n_tokens_content n_unique_tokens n_non_stop_words num_hrefs
## 1             12              219       0.6635945                1         4
## 2              9              255       0.6047431                1         3
## 3              9              211       0.5751295                1         3
## 4              9              531       0.5037879                1         9
## 5             13             1072       0.4156456                1        19
## 6             10              370       0.5598886                1         2
##   num_self_hrefs num_imgs num_videos average_token_length num_keywords
## 1              2        1          0             4.680365            5
## 2              1        1          0             4.913725            4
## 3              1        1          0             4.393365            6
## 4              0        1          0             4.404896            7
## 5             19       20          0             4.682836            7
## 6              2        0          0             4.359459            9
##   kw_max_max global_sentiment_polarity avg_positive_polarity title_subjectivity
## 1          0                0.09256198             0.3786364          0.5000000
## 2          0                0.14894781             0.2869146          0.0000000
## 3          0                0.32333333             0.4958333          0.0000000
## 4          0                0.10070467             0.3859652          0.0000000
## 5          0                0.28100348             0.4111274          0.4545455
## 6          0                0.07118419             0.3506100          0.6428571
##   title_sentiment_polarity abs_title_subjectivity abs_title_sentiment_polarity
## 1               -0.1875000             0.00000000                    0.1875000
## 2                0.0000000             0.50000000                    0.0000000
## 3                0.0000000             0.50000000                    0.0000000
## 4                0.0000000             0.50000000                    0.0000000
## 5                0.1363636             0.04545455                    0.1363636
## 6                0.2142857             0.14285714                    0.2142857
##   shares fav
## 1    593  NO
## 2    711  NO
## 3   1500 YES
## 4   1200  NO
```

```
## 5    505  NO
## 6    855  NO
```

```
set.seed(12345)
news_p_rand <- news_p[order(runif(10000)), ]
news_ptrain <- news_p_rand[1:9000, ]
news_ptest <- news_p_rand[9001:10000, ]
prop.table(table(news_ptrain$fav))
```

```
##
##        NO       YES
## 0.4308889 0.5691111
```

```
prop.table(table(news_ptest$fav))
```

```
##
##    NO   YES
## 0.414 0.586
```

```
library(C50)
newsp_model <- C5.0(x = news_ptrain[,c(-19,-18)], y = as.factor(news_ptrain$fav))
summary(newsp_model)
```

```
##
## Call:
## C5.0.default(x = news_ptrain[, c(-19, -18)], y = as.factor(news_ptrain$fav))
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Oct 18 19:40:42 2020
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 9000 cases (18 attributes) from undefined.data
##
## Decision tree:
##
## n_unique_tokens <= 0.4466737:
## :...kw_max_max <= 17100:
## :   :...n_tokens_content <= 1215: NO (29/5)
## :   :   n_tokens_content > 1215: YES (8/2)
## :   kw_max_max > 17100:
## :   :...kw_max_max <= 617900:
## :       :...n_tokens_title <= 10: YES (426/99)
## :       :   n_tokens_title > 10:
## :       :   :...num_hrefs <= 27: YES (176/57)
## :       :       num_hrefs > 27:
## :       :       :...num_hrefs <= 62: NO (11/1)
## :       :           num_hrefs > 62: YES (2)
## :       kw_max_max > 617900:
## :       :...num_self_hrefs > 0: YES (427/136)
## :           num_self_hrefs <= 0:
```

```
## :           :...num_keywords > 8:
## :               :...n_non_stop_words <= 0: YES (7/1)
## :               :   n_non_stop_words > 0: NO (32/7)
## :               num_keywords <= 8:
## :               :...abs_title_subjectivity <= 0.4166667:
## :                   :...n_tokens_title <= 11: NO (33/10)
## :                   :   n_tokens_title > 11: YES (5)
## :                   abs_title_subjectivity > 0.4166667:
## :                   :...n_tokens_title <= 6: NO (2)
## :                       n_tokens_title > 6: YES (35/5)
## n_unique_tokens > 0.4466737:
## :...kw_max_max <= 617900:
##     :...kw_max_max > 80400: YES (1832/633)
##     :   kw_max_max <= 80400:
##     :   :...num_self_hrefs <= 4:
##     :       :...abs_title_sentiment_polarity > 0.5125:
##     :       :   :...num_imgs > 12: YES (11)
##     :       :   :   num_imgs <= 12:
##     :       :   :   :...average_token_length <= 4.946988: YES (114/31)
##     :       :   :       average_token_length > 4.946988: NO (21/7)
##     :       :   abs_title_sentiment_polarity <= 0.5125:
##     :       :   :...kw_max_max > 39400: YES (1578/690)
##     :       :       kw_max_max <= 39400:
##     :       :       :...num_self_hrefs > 2:
##     :       :           :...global_sentiment_polarity <= 0.1456514: YES (52/14)
##     :       :           :   global_sentiment_polarity > 0.1456514: NO (52/21)
##     :       :           num_self_hrefs <= 2:
##     :       :           :...num_videos <= 0: NO (195/92)
##     :       :               num_videos > 0:
##     :       :               :...title_sentiment_polarity <= 0.075: NO (28/3)
##     :       :                   title_sentiment_polarity > 0.075:
##     :       :                   :...kw_max_max > 37400: YES (5)
##     :       :                       kw_max_max <= 37400:
##     :       :                       :...kw_max_max <= 17100: YES (3)
##     :       :                           kw_max_max > 17100: NO (8/2)
##     :       num_self_hrefs > 4:
##     :       :...global_sentiment_polarity > 0.2588357: NO (25/3)
##     :           global_sentiment_polarity <= 0.2588357:
##     :           :...avg_positive_polarity <= 0.3272109: NO (88/30)
##     :               avg_positive_polarity > 0.3272109:
##     :               :...num_imgs <= 3:
##     :                   :...num_keywords <= 7: NO (75/30)
##     :                   :   num_keywords > 7: YES (81/33)
##     :                   num_imgs > 3:
##     :                   :...num_videos > 1: NO (3)
##     :                       num_videos <= 1:
##     :                       :...avg_positive_polarity <= 0.3465233: YES (12)
##     :                           avg_positive_polarity > 0.3465233:
##     :                           :...abs_title_subjectivity > 0.4166667: YES (27/4)
##     :                               abs_title_subjectivity <= 0.4166667:
##     :                               :...n_tokens_content <= 813: NO (10/2)
##     :                                   n_tokens_content > 813: YES (5)
##     kw_max_max > 617900:
##     :...num_hrefs > 13:
```

```
##         :...num_self_hrefs <= 0: NO (79/31)
##         :   num_self_hrefs > 0:
##         :   :...n_tokens_title > 9:
##         :       :...average_token_length <= 4.892193: YES (209/78)
##         :       :   average_token_length > 4.892193: NO (82/29)
##         :       n_tokens_title <= 9:
##         :       :...kw_max_max > 690400: YES (60/20)
##         :           kw_max_max <= 690400:
##         :           :...num_hrefs > 44: YES (13)
##         :               num_hrefs <= 44:
##         :               :...num_hrefs <= 34: YES (213/52)
##         :                   num_hrefs > 34:
##         :                   :...avg_positive_polarity <= 0.4902552: NO (16/4)
##         :                       avg_positive_polarity > 0.4902552: YES (4)
##         num_hrefs <= 13:
##         :...num_imgs <= 0:
##             :...title_sentiment_polarity <= -0.025:
##             :   :...num_keywords > 7:
##             :   :   :...n_tokens_content <= 83: YES (5)
##             :   :   :   n_tokens_content > 83: NO (83/17)
##             :   :   num_keywords <= 7:
##             :   :   :...avg_positive_polarity <= 0.3493939:
##             :   :       :...kw_max_max <= 690400: YES (32/5)
##             :   :       :   kw_max_max > 690400: NO (2)
##             :   :       avg_positive_polarity > 0.3493939:
##             :   :       :...abs_title_subjectivity <= 0.02222222: YES (3)
##             :   :           abs_title_subjectivity > 0.02222222: NO (46/14)
##             :   title_sentiment_polarity > -0.025:
##             :   :...global_sentiment_polarity > 0.002449495: YES (651/253)
##             :       global_sentiment_polarity <= 0.002449495:
##             :       :...kw_max_max > 690400:
##             :           :...title_sentiment_polarity <= 0.06818182: NO (3)
##             :           :   title_sentiment_polarity > 0.06818182: YES (3)
##             :           kw_max_max <= 690400:
##             :           :...global_sentiment_polarity > -0.006586199: NO (9)
##             :               global_sentiment_polarity <= -0.006586199:
##             :               :...abs_title_subjectivity <= 0.125: NO (7/1)
##             :                   abs_title_subjectivity > 0.125:
##             :                   :...n_unique_tokens <= 0.6138614: YES (9)
##             :                       n_unique_tokens > 0.6138614:
##             :                       :...n_tokens_title <= 9: NO (20/5)
##             :                           n_tokens_title > 9: [S1]
##             num_imgs > 0:
##             :...title_sentiment_polarity > 0.7: YES (30/6)
##                 title_sentiment_polarity <= 0.7:
##                 :...kw_max_max > 690400: NO (225/80)
##                     kw_max_max <= 690400:
##                     :...num_videos > 3:
##                         :...num_keywords <= 5: NO (5/1)
##                         :   num_keywords > 5: YES (39/8)
##                         num_videos <= 3:
##                         :...n_tokens_title <= 6: YES (71/27)
##                             n_tokens_title > 6:
##                             :...average_token_length <= 4.408367:
```

```
##                                 :...n_tokens_title > 11:
##                                 :   :...num_hrefs <= 4: YES (29/11)
##                                 :   :   num_hrefs > 4:
##                                 :   :   :...title_subjectivity <= 0.8: NO (25/3)
##                                 :   :       title_subjectivity > 0.8: YES (3)
##                                 :   n_tokens_title <= 11: [S2]
##                                 average_token_length > 4.408367:
##                                 :...num_self_hrefs > 2: NO (661/252)
##                                     num_self_hrefs <= 2:
##                                     :...num_imgs > 7:
##                                         :...n_unique_tokens <= 0.4776786: NO (6)
##                                         :   n_unique_tokens > 0.4776786: YES (26/5)
##                                         num_imgs <= 7:
##                                         :...num_videos > 0: NO (95/31)
##                                             num_videos <= 0:
##                                             :...num_keywords > 9: NO (64/19)
##                                                 num_keywords <= 9: [S3]
##
## SubTree [S1]
##
## average_token_length <= 4.461285: NO (4/1)
## average_token_length > 4.461285: YES (8)
##
## SubTree [S2]
##
## global_sentiment_polarity > 0.1577778: YES (63/13)
## global_sentiment_polarity <= 0.1577778:
## :...num_videos <= 0:
##     :...average_token_length > 4.400285: YES (6)
##     :   average_token_length <= 4.400285:
##     :   :...avg_positive_polarity <= 0.3125:
##     :       :...num_keywords <= 8: YES (17/3)
##     :       :   num_keywords > 8: NO (3)
##     :       avg_positive_polarity > 0.3125:
##     :       :...n_tokens_title <= 7: YES (4/1)
##     :           n_tokens_title > 7: NO (46/13)
##     num_videos > 0:
##     :...num_videos > 1: NO (2)
##         num_videos <= 1:
##         :...average_token_length > 4.31361: YES (13/1)
##             average_token_length <= 4.31361:
##             :...global_sentiment_polarity <= 0.09637173: NO (7)
##                 global_sentiment_polarity > 0.09637173: YES (4)
##
## SubTree [S3]
##
## num_self_hrefs > 1: NO (203/87)
## num_self_hrefs <= 1:
## :...num_self_hrefs <= 0:
##     :...n_tokens_content <= 500: NO (137/60)
##     :   n_tokens_content > 500:
##     :   :...title_sentiment_polarity > 0.13:
##     :       :...abs_title_subjectivity <= 0.3: NO (12/3)
##     :       :   abs_title_subjectivity > 0.3: YES (5/1)
```

```
##     :       title_sentiment_polarity <= 0.13:
##     :       :...n_unique_tokens > 0.4814815: YES (54/15)
##     :           n_unique_tokens <= 0.4814815:
##     :           :...global_sentiment_polarity <= 0.09708565: YES (4)
##     :               global_sentiment_polarity > 0.09708565: NO (10/3)
##     num_self_hrefs > 0:
##     :...n_tokens_title <= 11: NO (129/59)
##         n_tokens_title > 11:
##         :...n_tokens_content > 253: YES (12)
##             n_tokens_content <= 253:
##             :...num_keywords > 6: YES (2)
##                 num_keywords <= 6:
##                 :...global_sentiment_polarity <= -0.04444445: YES (2)
##                     global_sentiment_polarity > -0.04444445: NO (7)
##
##
## Evaluation on training data (9000 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      92 3130(34.8%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##    1674  2204    (a): class NO
##     926  4196    (b): class YES
##
##
##  Attribute usage:
##
##  100.00% n_unique_tokens
##  100.00% kw_max_max
##   55.99% num_self_hrefs
##   41.90% num_hrefs
##   36.28% num_imgs
##   33.79% n_tokens_title
##   32.78% title_sentiment_polarity
##   23.21% average_token_length
##   22.97% abs_title_sentiment_polarity
##   22.91% num_videos
##   14.80% global_sentiment_polarity
##   12.73% num_keywords
##    5.27% avg_positive_polarity
##    4.28% n_tokens_content
##    2.57% abs_title_subjectivity
##    0.43% n_non_stop_words
##    0.31% title_subjectivity
##
##
## Time: 0.1 secs
```

```r
fav_pred <- predict(newsp_model, news_ptest)
library(gmodels)
CrossTable(news_ptest$fav, fav_pred, prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE, dnn = c( 'Actua
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                 | Predicted Favorite
## Actual Favorite |        NO |       YES | Row Total |
## ----------------|-----------|-----------|-----------|
##              NO |       127 |       287 |       414 |
##                 |     0.127 |     0.287 |           |
## ----------------|-----------|-----------|-----------|
##             YES |       130 |       456 |       586 |
##                 |     0.130 |     0.456 |           |
## ----------------|-----------|-----------|-----------|
##    Column Total |       257 |       743 |      1000 |
## ----------------|-----------|-----------|-----------|
##
##
```

It can be seen that 59% accuracy is with the above model. Let's implement another model.

```r
library(randomForest)

news_p_random_forest_model<- randomForest(as.factor(fav)~.,data=news_ptrain[,-18])
summary(news_p_random_forest_model)
```

```
##                 Length Class  Mode
## call                3  -none- call
## type                1  -none- character
## predicted        9000  factor numeric
## err.rate         1500  -none- numeric
## confusion           6  -none- numeric
## votes           18000  matrix numeric
## oob.times        9000  -none- numeric
## classes             2  -none- character
## importance         17  -none- numeric
## importanceSD        0  -none- NULL
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
```

```
## forest             14  -none- list
## y                9000  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
fac_pred_rf <- predict(news_p_random_forest_model, news_ptest)
(p <- table(fac_pred_rf, news_ptest$fav))
```

```
##
## fac_pred_rf  NO YES
##         NO  142 118
##         YES 272 468
```

```
(Accuracy <- sum(diag(p))/sum(p)*100)
```

```
## [1] 61
```

```
importance(news_p_random_forest_model)
```

```
##                              MeanDecreaseGini
## n_tokens_title                       219.2460
## n_tokens_content                     384.5708
## n_unique_tokens                      433.0941
## n_non_stop_words                     374.2535
## num_hrefs                            281.9677
## num_self_hrefs                       214.9019
## num_imgs                             133.8561
## num_videos                            93.2574
## average_token_length                 439.2662
## num_keywords                         206.9241
## kw_max_max                           180.7170
## global_sentiment_polarity            429.9007
## avg_positive_polarity                413.7598
## title_subjectivity                   154.1838
## title_sentiment_polarity             168.4176
## abs_title_subjectivity               139.2304
## abs_title_sentiment_polarity         138.4482
```

From the above, it can be seen that by using random forest, an accuracy of 60.5% is achieved that is relatively higher than that of the previous Tree based classification model.

Summary:

Upon implementing both Decision Tree and Random Forest algorithm approaches on the News Popularity dataset to predict if a certain news is a favorite among and to understand the share in the market, a conclusion can be made that both the models have obtained similar results in terms of accuracy. The Tree based classification model has an accuracy of 59% whereas the Random Forest Model has an accuracy of 60.5%.