IB031 - Project

Popis datasetu

Dataset, který jsem si zvolil klasifikuje bílá a červená vína podle jejich kvality na základě atributů: kyselost, cukernatost, obsah síry, hustota, pH a obsah alkoholu. Dataset obsahuje celkem 6497 vín. ## Nahrání dat a knihoven

```
library(RWeka)

wine.all <- read.csv("winequalityN.csv")
```

Explorační analýza

Dataset obsahuje celkem 6497 vín, z toho je 1599 červených a 4898 bílých. 7 z 13 atributů obsahují nějaké chybějící hodnoty.

```
head(wine.all)
```

```
##    type fixed.acidity volatile.acidity citric.acid residual.sugar
## 1 white           7.0             0.27        0.36           20.7
## 2 white           6.3             0.30        0.34            1.6
## 3 white           8.1             0.28        0.40            6.9
## 4 white           7.2             0.23        0.32            8.5
## 5 white           7.2             0.23        0.32            8.5
## 6 white           8.1             0.28        0.40            6.9
##   chlorides free.sulfur.dioxide total.sulfur.dioxide density   pH
## 1     0.045                  45                  170  1.0010 3.00
## 2     0.049                  14                  132  0.9940 3.30
## 3     0.050                  30                   97  0.9951 3.26
## 4     0.058                  47                  186  0.9956 3.19
## 5     0.058                  47                  186  0.9956 3.19
## 6     0.050                  30                   97  0.9951 3.26
##   sulphates alcohol quality
## 1      0.45     8.8       6
## 2      0.49     9.5       6
## 3      0.44    10.1       6
## 4      0.40     9.9       6
## 5      0.40     9.9       6
## 6      0.44    10.1       6
```

```
tail(wine.all)
```

```
##      type fixed.acidity volatile.acidity citric.acid residual.sugar
## 6492  red           6.8            0.620        0.08            1.9
## 6493  red           6.2            0.600        0.08            2.0
## 6494  red           5.9            0.550        0.10            2.2
## 6495  red           6.3            0.510        0.13            2.3
## 6496  red           5.9            0.645        0.12            2.0
## 6497  red           6.0            0.310        0.47            3.6
##      chlorides free.sulfur.dioxide total.sulfur.dioxide density   pH
## 6492     0.068                  28                   38 0.99651 3.42
## 6493     0.090                  32                   44 0.99490 3.45
## 6494     0.062                  39                   51 0.99512 3.52
## 6495     0.076                  29                   40 0.99574 3.42
```

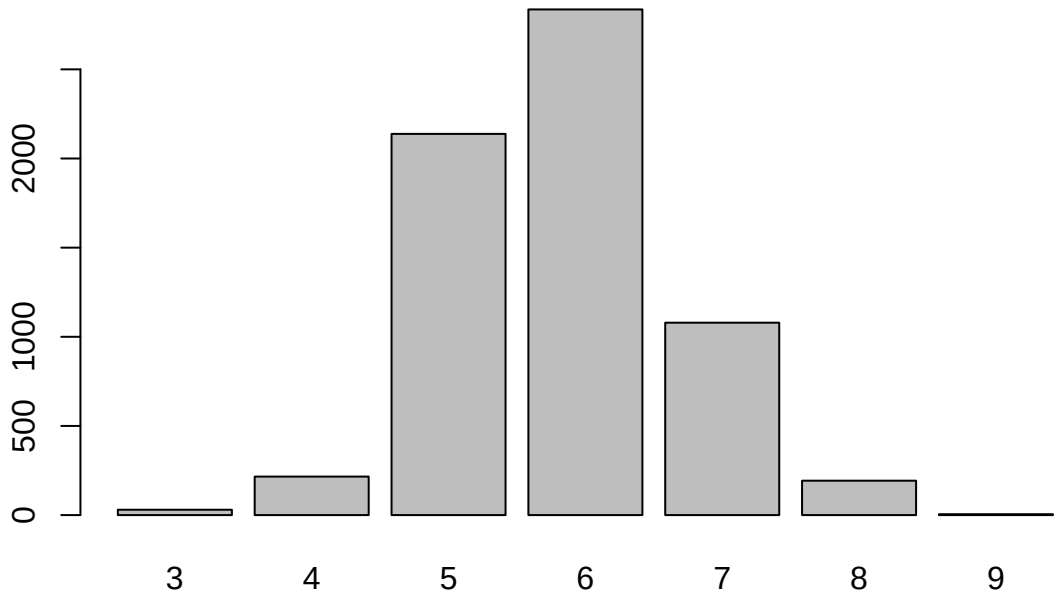
```
## 6496     0.075                 32                  44 0.99547 3.57
## 6497     0.067                 18                  42 0.99549 3.39
##      sulphates alcohol quality
## 6492      0.82     9.5       6
## 6493      0.58    10.5       5
## 6494        NA    11.2       6
## 6495      0.75    11.0       6
## 6496      0.71    10.2       5
## 6497      0.66    11.0       6
```

```
summary(wine.all)
```

```
##     type       fixed.acidity    volatile.acidity  citric.acid
##  red  :1599   Min.   : 3.800   Min.   :0.0800   Min.   :0.0000
##  white:4898   1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500
##               Median : 7.000   Median :0.2900   Median :0.3100
##               Mean   : 7.217   Mean   :0.3397   Mean   :0.3187
##               3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900
##               Max.   :15.900   Max.   :1.5800   Max.   :1.6600
##               NA's   :10       NA's   :8        NA's   :3
##  residual.sugar     chlorides       free.sulfur.dioxide
##  Min.   : 0.600   Min.   :0.00900   Min.   :  1.00
##  1st Qu.: 1.800   1st Qu.:0.03800   1st Qu.: 17.00
##  Median : 3.000   Median :0.04700   Median : 29.00
##  Mean   : 5.444   Mean   :0.05604   Mean   : 30.53
##  3rd Qu.: 8.100   3rd Qu.:0.06500   3rd Qu.: 41.00
##  Max.   :65.800   Max.   :0.61100   Max.   :289.00
##  NA's   :2        NA's   :2
##  total.sulfur.dioxide    density             pH          sulphates
##  Min.   :  6.0        Min.   :0.9871   Min.   :2.720   Min.   :0.2200
##  1st Qu.: 77.0        1st Qu.:0.9923   1st Qu.:3.110   1st Qu.:0.4300
##  Median :118.0        Median :0.9949   Median :3.210   Median :0.5100
##  Mean   :115.7        Mean   :0.9947   Mean   :3.218   Mean   :0.5312
##  3rd Qu.:156.0        3rd Qu.:0.9970   3rd Qu.:3.320   3rd Qu.:0.6000
##  Max.   :440.0        Max.   :1.0390   Max.   :4.010   Max.   :2.0000
##                                        NA's   :9       NA's   :4
##     alcohol         quality
##  Min.   : 8.00   Min.   :3.000
##  1st Qu.: 9.50   1st Qu.:5.000
##  Median :10.30   Median :6.000
##  Mean   :10.49   Mean   :5.818
##  3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :14.90   Max.   :9.000
##
```

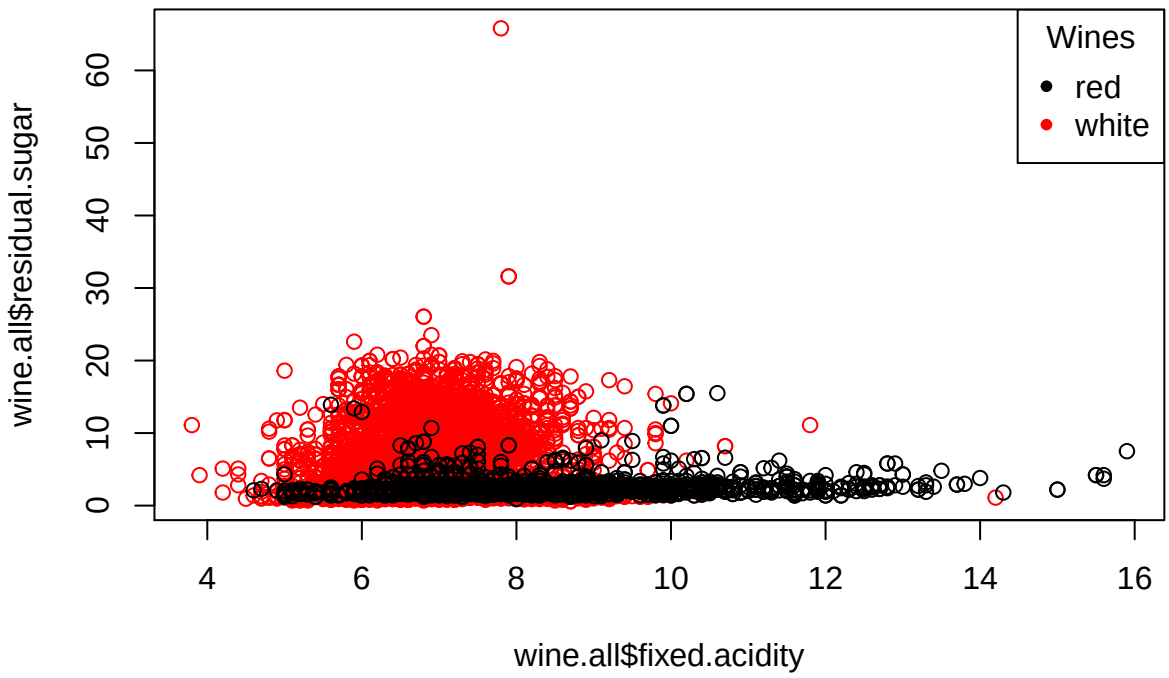
Zajímavé je rozložení hodnot atributu quality, která připomíná graf normálního rozložení.

```
wine.all$quality <- as.factor(wine.all$quality)
plot(wine.all$quality)
```



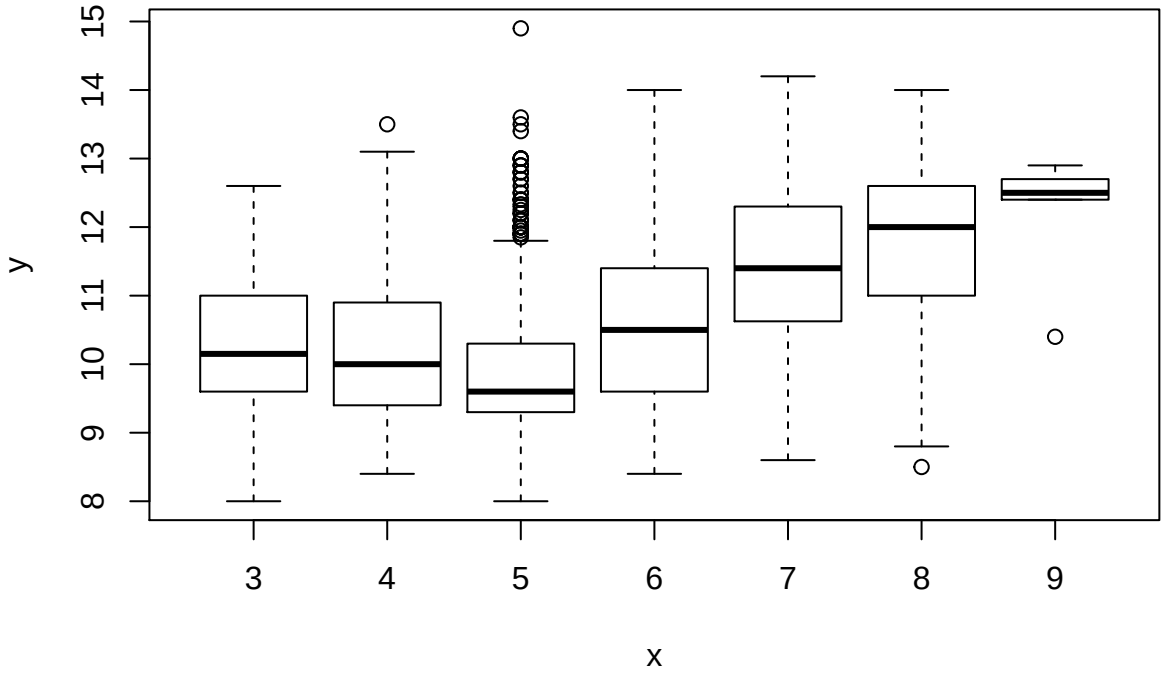
Nepřekvapivé je zjištění, že červená vína nejsou tak sladká jak bílé, a naopak bílé dosahují daleko nižší kyselosti.

```
plot(wine.all$fixed.acidity, wine.all$residual.sugar, col = wine.all$type)
legend("topright", levels(wine.all$type), col = 1:2, pch = 20, title = "Wines")
```



Jde také vidět že kvalitnější vína mají obvykle vyšší obsah alkoholu.

```
plot(wine.all$quality, wine.all$alcohol)
```



Baseline model

Jako první jsem se pokusil natrénovat model bez jakéhokoliv předzpracování. Výseldek dopadl nevalně, přesnost byla bohužel jen něco přes 33%.

```
wine.all$quality <- as.factor(wine.all$quality)
wine.train <- wine.all[1:3248, ]
wine.test <- wine.all[3249:6497, ]

### raw model without any preprocessing and tuning
model.wine.raw <- J48(quality ~ ., data = wine.train)
prediction.wine.raw <- predict(model.wine.raw, newdata = wine.test)
references.wine.raw <- wine.test$quality

confmat.wine.raw <- table(prediction.wine.raw, references.wine.raw)
accuracy.wine.raw <- sum(diag(confmat.wine.raw)) / sum(confmat.wine.raw)
accuracy.wine.raw
```

```
## [1] 0.3459526
```

Předzpracování

Je nutné změnit cílovou třídu na factor. Dále je třeba se vypořádat s chybějícími hodnotami, vzhledem k tomu, že všechny atributy kde se chybějící hodnoty nacházejí jsou číselné, rozhodl jsem se k nahrazení chybějících hodnot hodnotou průměrnou. Dále jsem všechny vína z kategorií kvality 3 a 9 přesunul do kategorií 4 resp. 8, jelikož těchto vín bylo velmi málo a negativně tyto položky ovlivňovaly přesnost modelu. Jako poslední jsem data náhodně promíchal a rozdělil na poloviny na trénovací a testovací množinu.

```r
## removing missing values

wine.all[is.na(wine.all$fixed.acidity), "fixed.acidity"] <- mean(wine.all$fixed.acidity, na.rm = T)
wine.all[is.na(wine.all$volatile.acidity), "volatile.acidity"] <- mean(wine.all$volatile.acidity, na.rm
wine.all[is.na(wine.all$citric.acid), "citric.acid"] <- mean(wine.all$citric.acid, na.rm = T)
wine.all[is.na(wine.all$residual.sugar), "residual.sugar"] <- mean(wine.all$residual.sugar, na.rm = T)
wine.all[is.na(wine.all$chlorides), "chlorides"] <- mean(wine.all$chlorides, na.rm = T)
wine.all[is.na(wine.all$pH), "pH"] <- mean(wine.all$pH, na.rm = T)
wine.all[is.na(wine.all$sulphates), "sulphates"] <- mean(wine.all$sulphates, na.rm = T)

## merging category no.3 to no.4 and no.9 to no.8
wine.all[(wine.all$quality == 3), "quality"] <- 4
wine.all[(wine.all$quality == 9), "quality"] <- 8
wine.all$quality <- droplevels(wine.all$quality, exclude = c(3,9))

wine.all$quality <- as.factor(wine.all$quality)

##data shufling
wine.all <- wine.all[sample(nrow(wine.all)), ]

wine.train <- wine.all[1:((nrow(wine.all)) * 0.7), ]
wine.test <- wine.all[(nrow(wine.all)*0.7):nrow(wine.all), ]
```

Model C4.5

Na natrénování tohoto modelu jsem použil algoritmus C4.5, v jazyce R implementovaný v knihovně RWeka a také v knihovně caret. Já použil implementaci z knihovny RWeka, kde je tento algoritmus implementovaný funkcí "J48". Algoritmus C4.5 vychází ze staršího algoritmu ID3, a tento algoritmus dále rozšiřuje. Používá se ke klasifikaci a tvorbě rozhodovacích klasifikačních stromů. Princip jeho funkce je následující: Spočte informační zisk jednotlivých atributů tak, aby co nejlépe rozdělovali danou množinu. Tento atribut se poté umístí do daného uzlu, který rozhoduje podle atributu s největším informačním ziskem a rekurzivně se pokračuje na podmnožinách daných rozdělením na předchozím uzlu.

```r
model.wine <- J48(quality ~ ., data = wine.train, control = Weka_control(R = F, M = 1))
prediction.wine <- predict(model.wine, wine.test)
```

Vyhodnocení modelu

```r
references.wine <- wine.test$quality

confmat.wine <- table(prediction.wine, references.wine)
confmat.wine
```

```
##                references.wine
## prediction.wine   4   5   6   7   8
##               4  22  25  20   1   0
##               5  31 411 183  15   2
##               6  29 169 510 124  23
##               7   5  37 104 163  13
##               8   0   6  18  18  21
```

```r
accuracy.wine <- sum(diag(confmat.wine)) / sum(confmat.wine)
accuracy.wine
```

```
## [1] 0.5779487
```

Bohužel model se nepodařilo natrénovat na více než něco kolem 58%. Není to mnoho, je však třeba brát v úvahu několik věcí. Jednak hodnocení kvality je subjektivní záležitost, a nelze ji jednoznačně odhadnout. Druhou věcí je fakt, že naprostá většina chybných klasifikací probíhá pouze o jednu třídu, ať už výš nebo níž. Po zvážení tohoto faktu jsem mírně upravil výpočet přesnosti tak, aby se za správný odhad považovalo pokud je víno zařazeno do správné kategorie nebo nanejvýš o jednu kategorii vedle. S touto tolerancí již přesnost dosahuje zhruba 93%, je proto zřejmé, že většina chybných klasifikací je pouze o jednu třídu.

```
### evaluation with toleration +- 1 class
accuracy.wine.with.tolerance <- confmat.wine[1:1] + confmat.wine[1,2]
for(i in 2:4){
  for(j in (i-1):(i+1)){
    accuracy.wine.with.tolerance <- accuracy.wine.with.tolerance + confmat.wine[i,j]
  }
}
accuracy.wine.with.tolerance <- accuracy.wine.with.tolerance + confmat.wine[5,4] + confmat.wine[5,5]
accuracy.wine.with.tolerance <- accuracy.wine.with.tolerance / sum(confmat.wine)
accuracy.wine.with.tolerance
```

```
## [1] 0.92
```

Tento model se tedy při použití na tomto konkrétním datasetu příliš neosvědčil. Je to dáno pravděpodobně větším množstvím možných výsledných klasifikací mezi kterými nelze přesně rozhodnout na základě daných atributů. V ostatních použitých modelech dopadly výsledky lépe, byť ne o mnoho. V algoritmu Random Forest byla přesnost okolo 70% a při algoritmu ID3 se pohybovala kolem 65%.

Dataset mushrooms

Explorační analýza

Tento dataset obsahuje 8124 položek a rozhoduje zda je houba jedovatá či nikoliv. Velikost množin jedovatých a jedlých hub je téměř stejná, žádné atributy neobsahují chybějící hodnoty. Z tohoto důvodu nebyla nutná prakticky žádná úprava ani žádné parametry modelu aby se dosáhlo přesnosti téměř 100%.

```
####### loading data #######
library(RWeka)
mushrooms.all <- read.csv("mushrooms.csv")
######## analysis ########
head(mushrooms.all)
```

```
##   class cap.shape cap.surface cap.color bruises odor gill.attachment
## 1     p         x           s         n       t    p               f
## 2     e         x           s         y       t    a               f
## 3     e         b           s         w       t    l               f
## 4     p         x           y         w       t    p               f
## 5     e         x           s         g       f    n               f
## 6     e         x           y         y       t    a               f
##   gill.spacing gill.size gill.color stalk.shape stalk.root
## 1            c         n          k           e          e
## 2            c         b          k           e          c
## 3            c         b          n           e          c
## 4            c         n          n           e          e
## 5            w         b          k           t          e
## 6            c         b          n           e          c
##   stalk.surface.above.ring stalk.surface.below.ring stalk.color.above.ring
## 1                        s                        s                      w
## 2                        s                        s                      w
## 3                        s                        s                      w
```

```
## 4                        s                        s                       w
## 5                        s                        s                       w
## 6                        s                        s                       w
##   stalk.color.below.ring veil.type veil.color ring.number ring.type
## 1                      w         p          w           o         p
## 2                      w         p          w           o         p
## 3                      w         p          w           o         p
## 4                      w         p          w           o         p
## 5                      w         p          w           o         e
## 6                      w         p          w           o         p
##   spore.print.color population habitat
## 1                 k          s       u
## 2                 n          n       g
## 3                 n          n       m
## 4                 k          s       u
## 5                 n          a       g
## 6                 k          n       g
```

```
summary(mushrooms.all)
```

```
##  class    cap.shape cap.surface   cap.color    bruises       odor
##  e:4208   b: 452    f:2320      n      :2284   f:4748   n      :3528
##  p:3916   c:   4    g:   4      g      :1840   t:3376   f      :2160
##           f:3152    s:2556      e      :1500            s      : 576
##           k: 828    y:3244      y      :1072            y      : 576
##           s:  32                w      :1040            a      : 400
##           x:3656                b      : 168            l      : 400
##                                 (Other): 220            (Other): 484
##  gill.attachment gill.spacing gill.size   gill.color   stalk.shape
##  a: 210          c:6812       b:5612    b      :1728   e:3516
##  f:7914          w:1312       n:2512    p      :1492   t:4608
##                                         w      :1202
##                                         n      :1048
##                                         g      : 752
##                                         h      : 732
##                                         (Other):1170
##  stalk.root stalk.surface.above.ring stalk.surface.below.ring
##  ?:2480     f: 552                   f: 600
##  b:3776     k:2372                   k:2304
##  c: 556     s:5176                   s:4936
##  e:1120     y:  24                   y: 284
##  r: 192
##
##
##  stalk.color.above.ring stalk.color.below.ring veil.type veil.color
##  w      :4464           w      :4384           p:8124    n:  96
##  p      :1872           p      :1872                     o:  96
##  g      : 576           g      : 576                     w:7924
##  n      : 448           n      : 512                     y:   8
##  b      : 432           b      : 432
##  o      : 192           o      : 192
##  (Other): 140           (Other): 156
##  ring.number ring.type spore.print.color population habitat
##  n:  36      e:2776    w      :2388      a: 384     d:3148
##  o:7488      f:  48    n      :1968      c: 340     g:2148
```

```
##  t: 600      l:1296    k      :1872      n: 400     l: 832
##              n:  36    h      :1632      s:1248     m: 292
##              p:3968    r      :  72      v:4040     p:1144
##                        b      :  48      y:1712     u: 368
##                        (Other): 144                 w: 192
```

```
###### preprocessing ########
mushrooms.all <- mushrooms.all[sample(nrow(mushrooms.all)), ]
mushrooms.train <- mushrooms.all[1:4062, ]
mushrooms.test <- mushrooms.all[4063:8124, ]
##### model #####
model.mushrooms <- J48(class ~ ., mushrooms.train, control = Weka_control(R = T))
prediction.mushrooms <- predict(model.mushrooms, mushrooms.test)
references.mushrooms <- mushrooms.test$class
confmat.mushrooms <- table(prediction.mushrooms, references.mushrooms)
confmat.mushrooms
```

```
##                     references.mushrooms
## prediction.mushrooms    e    p
##                    e 2116    2
##                    p    0 1944
```

```
accuracy.mushrooms <- sum(diag(confmat.mushrooms)) / sum(confmat.mushrooms)
accuracy.mushrooms
```

```
## [1] 0.9995076
```

Dataset cars

```
######## loading data #######
library(RWeka)
cars.all <- read.csv("car.data")
names(cars.all) <- c("buying", "maint", "doors", "persons", "lug_boot", "safety", "class")
head(cars.all)
```

```
##   buying maint doors persons lug_boot safety class
## 1  vhigh vhigh     2       2    small    med unacc
## 2  vhigh vhigh     2       2    small   high unacc
## 3  vhigh vhigh     2       2      med    low unacc
## 4  vhigh vhigh     2       2      med    med unacc
## 5  vhigh vhigh     2       2      med   high unacc
## 6  vhigh vhigh     2       2      big    low unacc
```

```
summary(cars.all)
```

```
##    buying       maint        doors      persons      lug_boot    safety
##  high :432   high :432   2    :431   2    :575   big  :576   high:576
##  low  :432   low  :432   3    :432   4    :576   med  :576   low :575
##  med  :432   med  :432   4    :432   more:576   small:575   med :576
##  vhigh:431   vhigh:431   5more:432
##    class
##  acc  : 384
##  good :  69
##  unacc:1209
##  vgood:  65
```

```
cars.all <- cars.all[sample(nrow(cars.all)), ]
cars.train <- cars.all[1:(nrow(cars.all)*0.7), ]
cars.test <- cars.all[(nrow(cars.all)*0.7):nrow(cars.all), ]
model.cars <- J48(class ~ ., cars.train, control = Weka_control(R = T, M = 1))
prediction.cars <- predict(model.cars, cars.test)
references.cars <- cars.test$class
confmat <- table(prediction.cars, references.cars)
confmat
```

```
##                references.cars
## prediction.cars acc good unacc vgood
##           acc   108    5    11     1
##           good    1   10     1     5
##           unacc  17    0   336     0
##           vgood   4    7     2    11
```

```
accuracy.cars <- sum(diag(confmat)) / sum(confmat)
accuracy.cars
```

```
## [1] 0.8959538
```