White_classificationtree_ordinal.R

nebojsahrnjez

2021-12-03

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.1.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-3
```

```
library(coefplot)
library(ggfortify)
```

```
## Registered S3 methods overwritten by 'ggfortify':
##   method         from
##   autoplot.acf   useful
##   fortify.acf    useful
##   fortify.kmeans useful
##   fortify.ts     useful
```

```
library(readr)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(moments)
library(ggpubr)
library(ggrepel)
library(qqplotr)
```

```
##
## Attaching package: 'qqplotr'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ordinal)
```

```
##
## Attaching package: 'ordinal'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
library(rpart.plot)
library(rpartScore)
library(DMwR2)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

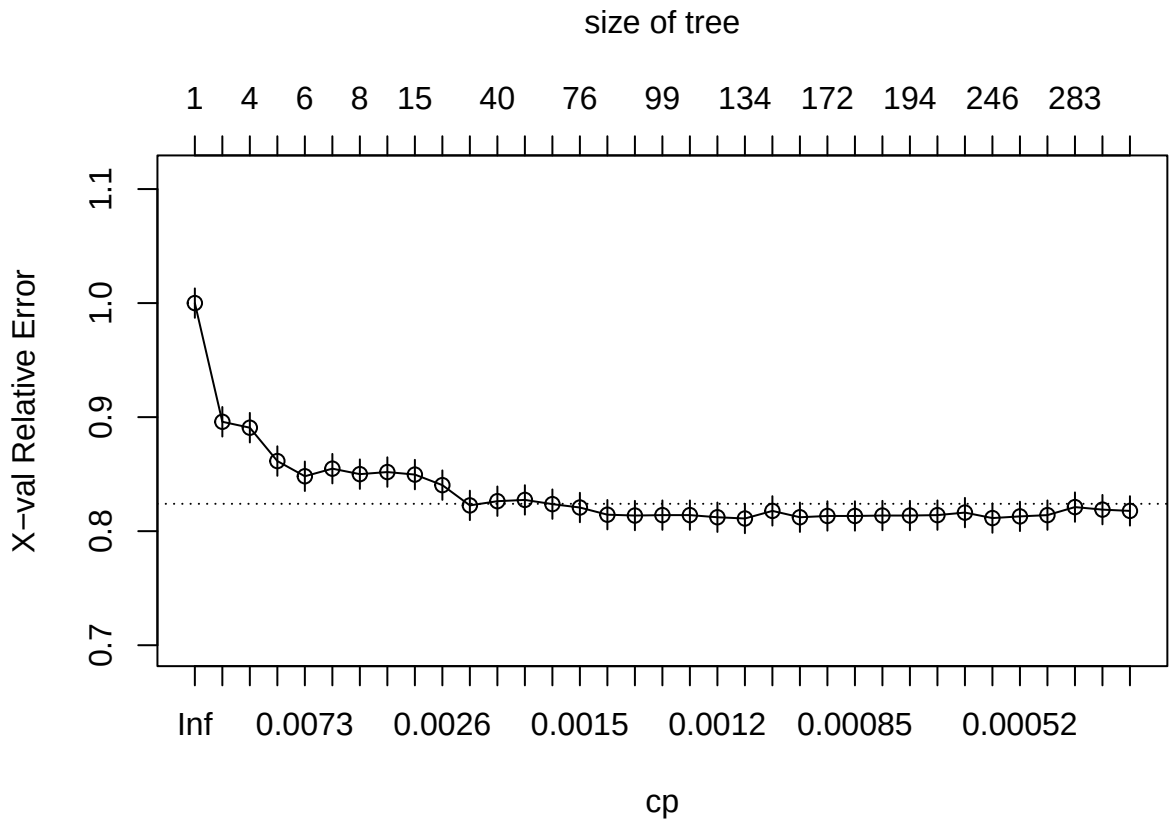
```
white <- read_csv("winequality-white.csv")
```

```
## Rows: 4898 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (12): fixed acidity, volatile acidity, citric acid, residual sugar, chlo...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sum(is.na(white))
```

```
## [1] 0
```

```
white <- na.omit(white)


dfw <- as.data.frame(white)

dfw$quality <- as.factor(dfw$quality)
names(dfw) <- make.names(names(dfw))

SEED <- 5864
set.seed(SEED)

test <- sample(1:nrow(dfw), size = nrow(dfw)/5)
train <- (-test)

dfw.train <- dfw[train,]
dfw.test <- dfw[test,]

#ordinaltree <- rpartScore(quality~., data = dfw.train)

#ordinaltree.predict <- predict(ordinaltree, newdata= dfw.test)

#mean(ordinaltree.predict==dfw.test$quality)
#Did not work so commented out

rpart.dfw <- rpart(quality~.,data=dfw,method ="class", cp=0.0000000000001)
plotcp(rpart.dfw)
```



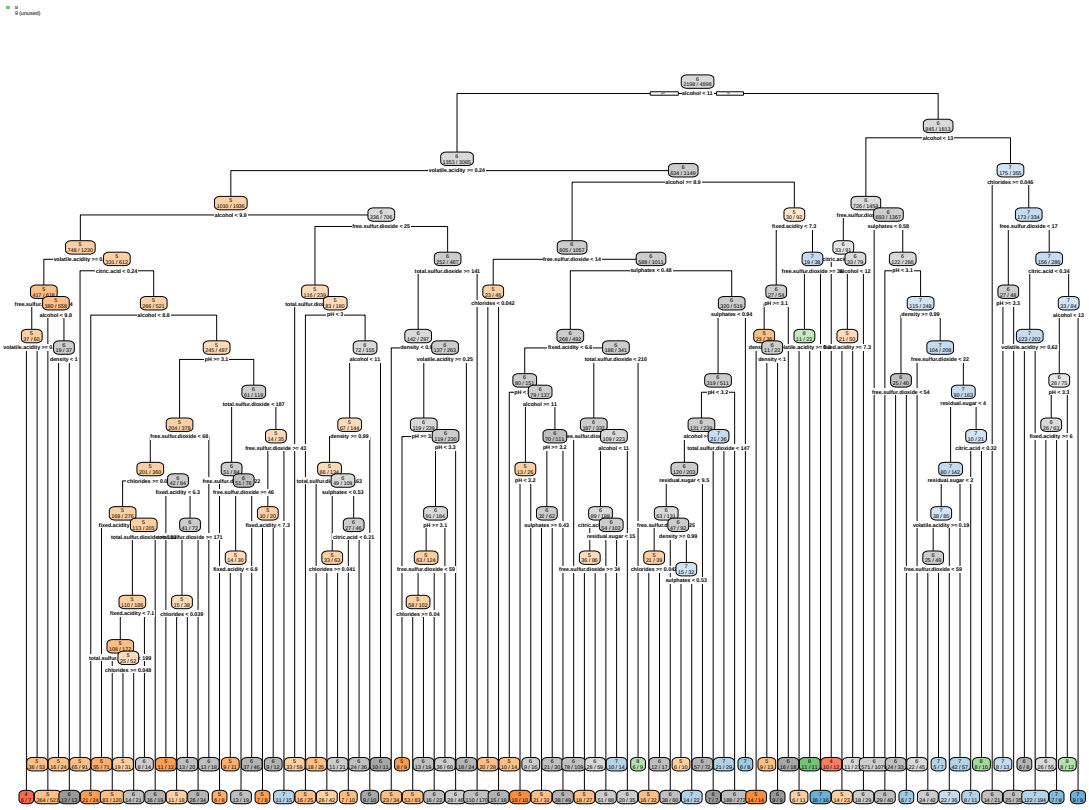

```
printcp(rpart.dfw)
```

```
##
## Classification tree:
## rpart(formula = quality ~ ., data = dfw, method = "class", cp = 1e-13)
##
## Variables actually used in tree construction:
##  [1] alcohol              chlorides            citric.acid
##  [4] density              fixed.acidity        free.sulfur.dioxide
##  [7] pH                   residual.sugar       sulphates
## [10] total.sulfur.dioxide volatile.acidity
##
## Root node error: 2700/4898 = 0.55125
##
## n= 4898
##
##            CP nsplit rel error  xerror     xstd
## 1  5.3889e-02      0   1.00000 1.00000 0.012892
## 2  2.7407e-02      2   0.89222 0.89593 0.012959
## 3  2.0741e-02      3   0.86481 0.89074 0.012958
## 4  1.1852e-02      4   0.84407 0.86148 0.012944
## 5  4.4444e-03      5   0.83222 0.84815 0.012933
## 6  3.7037e-03      6   0.82778 0.85481 0.012939
## 7  3.3333e-03      7   0.82407 0.85000 0.012935
## 8  3.2407e-03      9   0.81741 0.85185 0.012936
```

```
## 9  2.9630e-03     14   0.80000 0.84963 0.012934
## 10 2.2222e-03     21   0.77704 0.84037 0.012925
## 11 1.8519e-03     24   0.77037 0.82259 0.012904
## 12 1.6667e-03     39   0.73481 0.82630 0.012909
## 13 1.6296e-03     53   0.71037 0.82741 0.012910
## 14 1.5344e-03     58   0.70222 0.82370 0.012906
## 15 1.4815e-03     75   0.67296 0.82074 0.012902
## 16 1.3889e-03     88   0.64963 0.81444 0.012893
## 17 1.3333e-03     93   0.64259 0.81370 0.012892
## 18 1.2963e-03     98   0.63593 0.81407 0.012892
## 19 1.2346e-03    104   0.62815 0.81407 0.012892
## 20 1.2037e-03    126   0.58963 0.81222 0.012889
## 21 1.1111e-03    133   0.58037 0.81111 0.012888
## 22 9.8765e-04    155   0.55593 0.81778 0.012897
## 23 9.2593e-04    163   0.54778 0.81222 0.012889
## 24 8.6420e-04    171   0.54037 0.81333 0.012891
## 25 8.3333e-04    179   0.53259 0.81333 0.012891
## 26 8.0247e-04    185   0.52630 0.81370 0.012892
## 27 7.4074e-04    193   0.51889 0.81370 0.012892
## 28 6.4815e-04    237   0.48111 0.81407 0.012892
## 29 6.1728e-04    241   0.47852 0.81630 0.012895
## 30 5.5556e-04    245   0.47593 0.81148 0.012888
## 31 4.9383e-04    251   0.47259 0.81296 0.012890
## 32 3.7037e-04    254   0.47111 0.81407 0.012892
## 33 1.8519e-04    282   0.46074 0.82111 0.012902
## 34 1.2346e-04    289   0.45926 0.81889 0.012899
## 35 1.0000e-13    292   0.45889 0.81778 0.012897
```

```
#indicates a value around 100 has the lowest xerror and xstd, when we look
#its a value of tree of size 126, we can get this with a cp of 1.3333e-03

rpart.dfw.min <- prune(rpart.dfw, cp=1.3333e-03)

rpart.plot(rpart.dfw.min, extra = 2)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

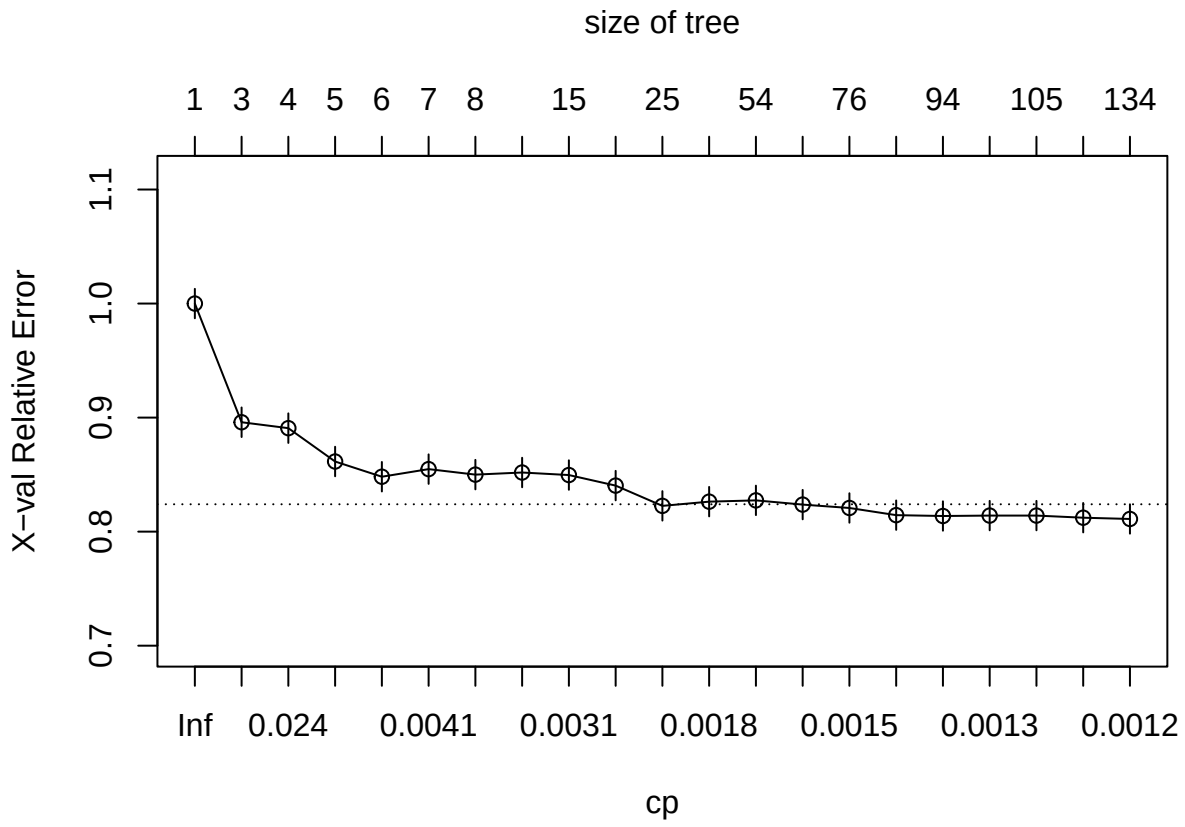


```
rpart.dfw.min2 <- rt.prune(rpart.dfw, se=0)
printcp(rpart.dfw.min2)
```

```
##
## Classification tree:
## rpart(formula = quality ~ ., data = dfw, method = "class", cp = 1e-13)
##
## Variables actually used in tree construction:
##  [1] alcohol              chlorides            citric.acid
##  [4] density              fixed.acidity        free.sulfur.dioxide
##  [7] pH                   residual.sugar       sulphates
## [10] total.sulfur.dioxide volatile.acidity
##
## Root node error: 2700/4898 = 0.55125
##
## n= 4898
##
##           CP nsplit rel error  xerror     xstd
## 1  0.0538889      0   1.00000 1.00000 0.012892
## 2  0.0274074      2   0.89222 0.89593 0.012959
## 3  0.0207407      3   0.86481 0.89074 0.012958
## 4  0.0118519      4   0.84407 0.86148 0.012944
## 5  0.0044444      5   0.83222 0.84815 0.012933
## 6  0.0037037      6   0.82778 0.85481 0.012939
## 7  0.0033333      7   0.82407 0.85000 0.012935
```

```
## 8  0.0032407      9   0.81741 0.85185 0.012936
## 9  0.0029630     14   0.80000 0.84963 0.012934
## 10 0.0022222     21   0.77704 0.84037 0.012925
## 11 0.0018519     24   0.77037 0.82259 0.012904
## 12 0.0016667     39   0.73481 0.82630 0.012909
## 13 0.0016296     53   0.71037 0.82741 0.012910
## 14 0.0015344     58   0.70222 0.82370 0.012906
## 15 0.0014815     75   0.67296 0.82074 0.012902
## 16 0.0013889     88   0.64963 0.81444 0.012893
## 17 0.0013333     93   0.64259 0.81370 0.012892
## 18 0.0012963     98   0.63593 0.81407 0.012892
## 19 0.0012346    104   0.62815 0.81407 0.012892
## 20 0.0012037    126   0.58963 0.81222 0.012889
## 21 0.0011111    133   0.58037 0.81111 0.012888
```

```
plotcp(rpart.dfw.min2)
```

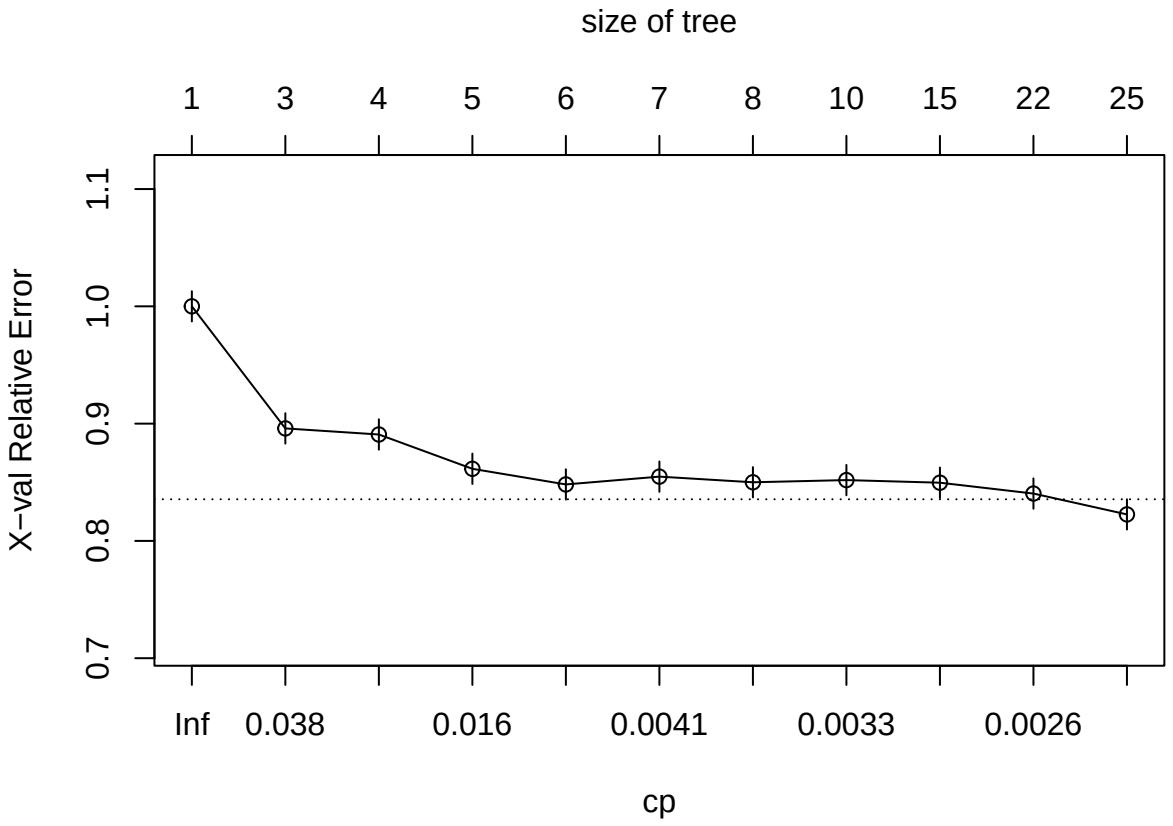



```
#Best value shown at tree size of 53 and a cp of 0.0015

rpart.dfw.1se <- rt.prune(rpart.dfw, se=1)
printcp(rpart.dfw.1se)
```

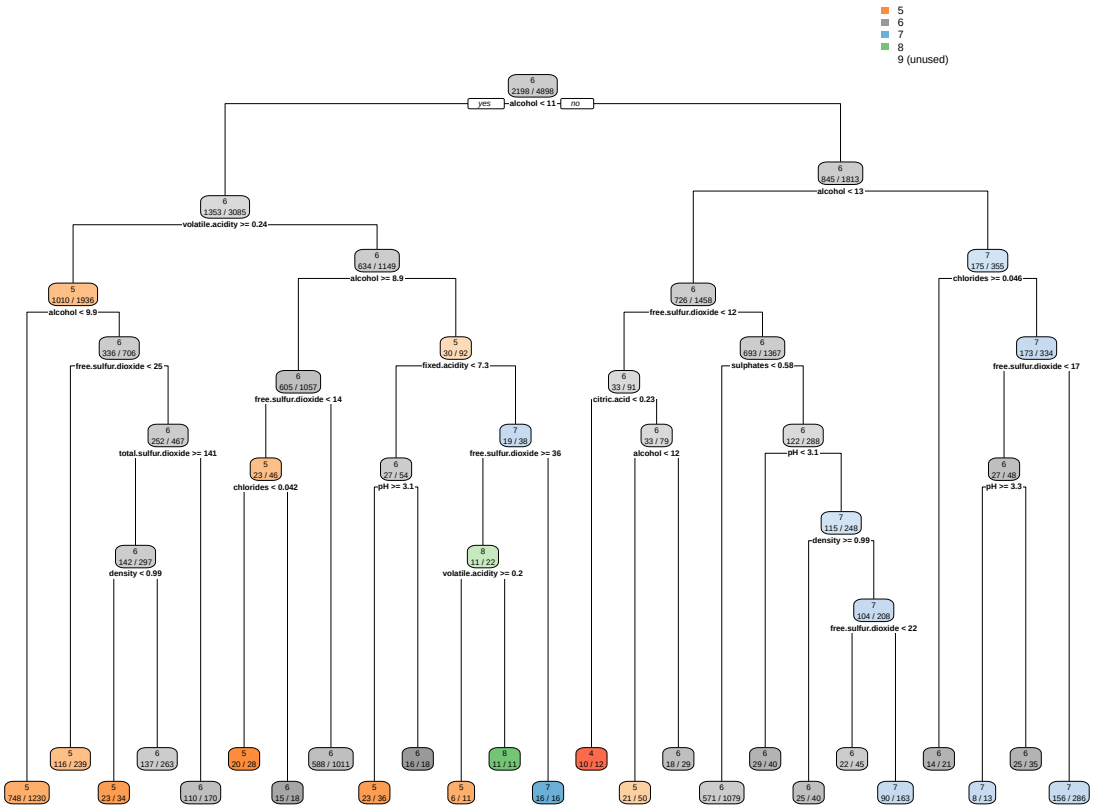
```
##
## Classification tree:
## rpart(formula = quality ~ ., data = dfw, method = "class", cp = 1e-13)
```

```
##
## Variables actually used in tree construction:
##  [1] alcohol              chlorides            citric.acid
##  [4] density              fixed.acidity        free.sulfur.dioxide
##  [7] pH                   sulphates            total.sulfur.dioxide
## [10] volatile.acidity
##
## Root node error: 2700/4898 = 0.55125
##
## n= 4898
##
##           CP nsplit rel error  xerror     xstd
## 1  0.0538889      0   1.00000 1.00000 0.012892
## 2  0.0274074      2   0.89222 0.89593 0.012959
## 3  0.0207407      3   0.86481 0.89074 0.012958
## 4  0.0118519      4   0.84407 0.86148 0.012944
## 5  0.0044444      5   0.83222 0.84815 0.012933
## 6  0.0037037      6   0.82778 0.85481 0.012939
## 7  0.0033333      7   0.82407 0.85000 0.012935
## 8  0.0032407      9   0.81741 0.85185 0.012936
## 9  0.0029630     14   0.80000 0.84963 0.012934
## 10 0.0022222     21   0.77704 0.84037 0.012925
## 11 0.0018519     24   0.77037 0.82259 0.012904
```

```
plotcp(rpart.dfw.1se)
```



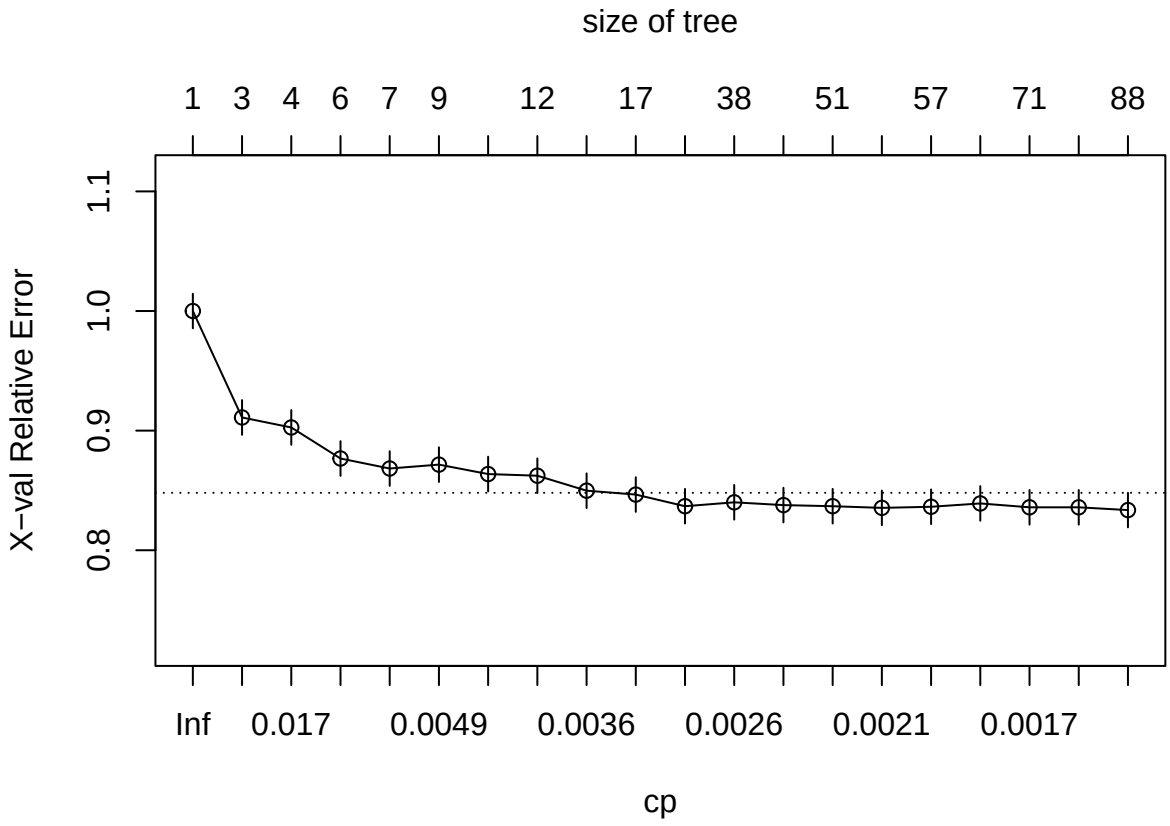
```
rpart.plot(rpart.dfw.1se, extra = 2)
```




```
test <- sample(1:nrow(dfw), size = nrow(dfw)/5)
train <- (-test)

dfw.train <- dfw[train,]
dfw.test <- dfw[test,]

rpart.dfw.train <- rpart(quality~., dfw.train, method = "class", cp = .0015)
rpart.dfw.pred <- predict(rpart.dfw.train, dfw.test, type = "class")

table(rpart.dfw.pred, dfw.test$quality)
```

```
##
## rpart.dfw.pred   3   4   5   6   7   8   9
##              3   0   0   0   0   0   0   0
##              4   0   1   0   2   0   0   0
##              5   2  12 182  78  10   1   0
##              6   2   7 131 298  84  17   0
##              7   0   0   4  53  70  14   0
##              8   0   0   0   5   3   3   0
##              9   0   0   0   0   0   0   0
```

```
plotcp(rpart.dfw.train)
```

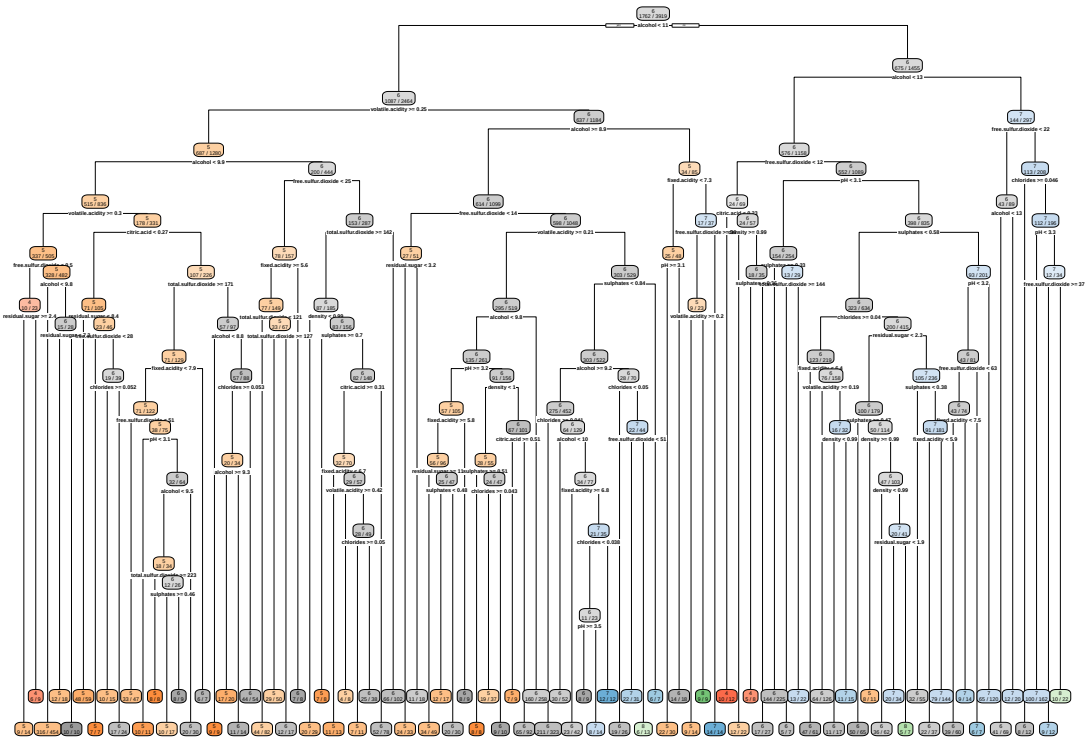



```
mean(rpart.dfw.pred==dfw.test$quality)
```

```
## [1] 0.5658836
```

```
rpart.plot(rpart.dfw.train, extra =2)
```

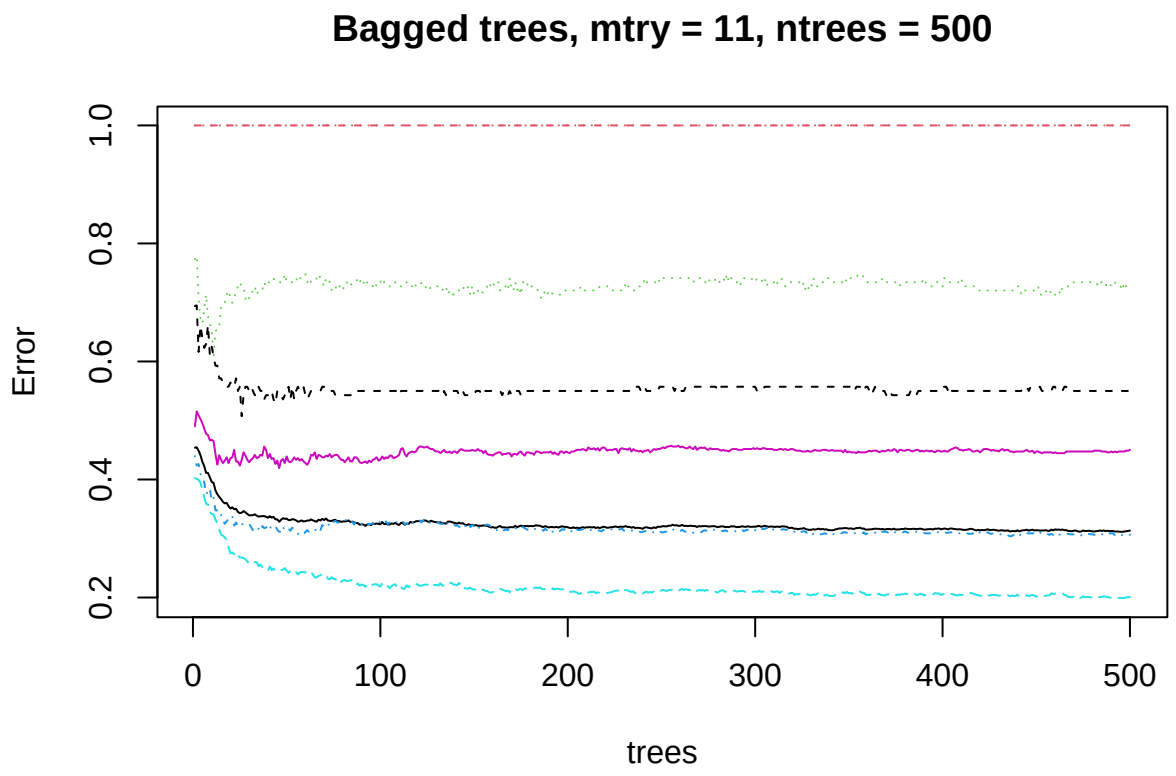
```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



```
#Training-test shows a classification rate of 53%

M <- 11


bag.dfw <- randomForest(quality~., data =dfw.train,mtry=11, importance = TRUE)

bag.dfw
```

```
##
## Call:
##  randomForest(formula = quality ~ ., data = dfw.train, mtry = 11,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 11
##
##         OOB estimate of  error rate: 31.33%
## Confusion matrix:
##   3  4   5    6   7  8 9 class.error
## 3 0  0   7    9   0  0 0   1.0000000
## 4 0 38  69   35   1  0 0   0.7342657
## 5 0 10 791  328  11  0 0   0.3061404
## 6 0  3 230 1408 117  4 0   0.2009081
## 7 0  1  13  302 392  5 0   0.4502104
## 8 0  0   2   39  37 62 0   0.5571429
## 9 0  0   0    3   2  0 0   1.0000000
```

```
plot(bag.dfw, main = "Bagged trees, mtry = 11, ntrees = 500")
```




```
bag.dfw.pred <- predict(bag.dfw, newdata = dfw.test)

mean(bag.dfw.pred==dfw.test$quality)
```

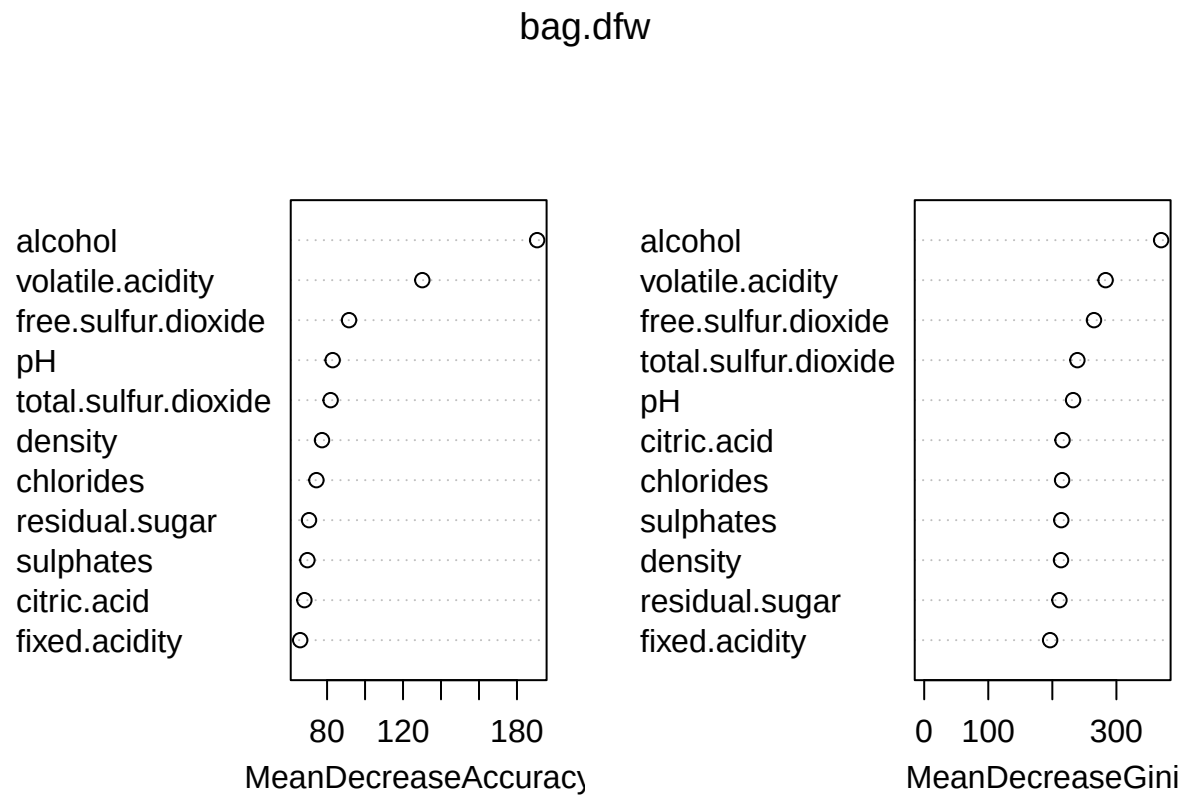
```
## [1] 0.6874362
```

```
importance(bag.dfw)
```

```
##                              3        4         5        6         7        8
## fixed.acidity        -0.4481590 13.37463  45.07895 39.53022  39.13927 33.25092
## volatile.acidity     -2.9248068 38.50298  82.08382 74.38900  83.02177 58.33533
## citric.acid           1.7312543 28.27217  40.72504 42.65822  44.35320 29.64371
## residual.sugar       -2.3934688 12.92130  44.37712 45.33576  40.06357 32.69743
## chlorides             1.6867048 20.48584  43.87126 32.58944  54.92489 29.11170
## free.sulfur.dioxide   0.5309707 37.41225  49.33654 54.99415  48.98204 39.39221
## total.sulfur.dioxide -0.3703086 17.23541  44.16990 37.62473  59.00534 33.66269
## density              -0.7279924 12.28133  41.68059 44.73538  42.41701 32.07052
## pH                   -0.8414521 14.64292  54.64017 42.77114  55.13248 32.85846
## sulphates            -2.2203159 16.29358  41.89140 44.10015  43.28365 38.38661
## alcohol              -4.3925636 27.20660 132.69672 61.00222 137.05327 90.54359
##                              9 MeanDecreaseAccuracy MeanDecreaseGini
## fixed.acidity        0.0000000             65.85026         196.5504
```

```
## volatile.acidity      1.9042342          130.15544        283.1526
## citric.acid           1.4812160           68.05778        215.8729
## residual.sugar        0.4473031           70.49796        211.1166
## chlorides             1.6713157           74.39008        215.2397
## free.sulfur.dioxide  -1.5109947           91.57363        265.0164
## total.sulfur.dioxide -1.3440623           81.87391        239.0935
## density               1.8966081           77.32856        213.5605
## pH                   -0.1561776           82.93654        232.4078
## sulphates             0.1084665           69.65340        213.9229
## alcohol               0.7750618          190.61404        369.7868
```

```
varImpPlot(bag.dfw)
```




```
#Training-test shows a classification of ~65% for bagged random forest

rf.dfw <- randomForest(quality~., data=dfw.train,mtry=5, importance = TRUE)
rf.dfw.pred <- predict(rf.dfw, newdata = dfw.test)

mean(rf.dfw.pred==dfw.test$quality)
```

```
## [1] 0.6874362
```

```
rf.dfw
```

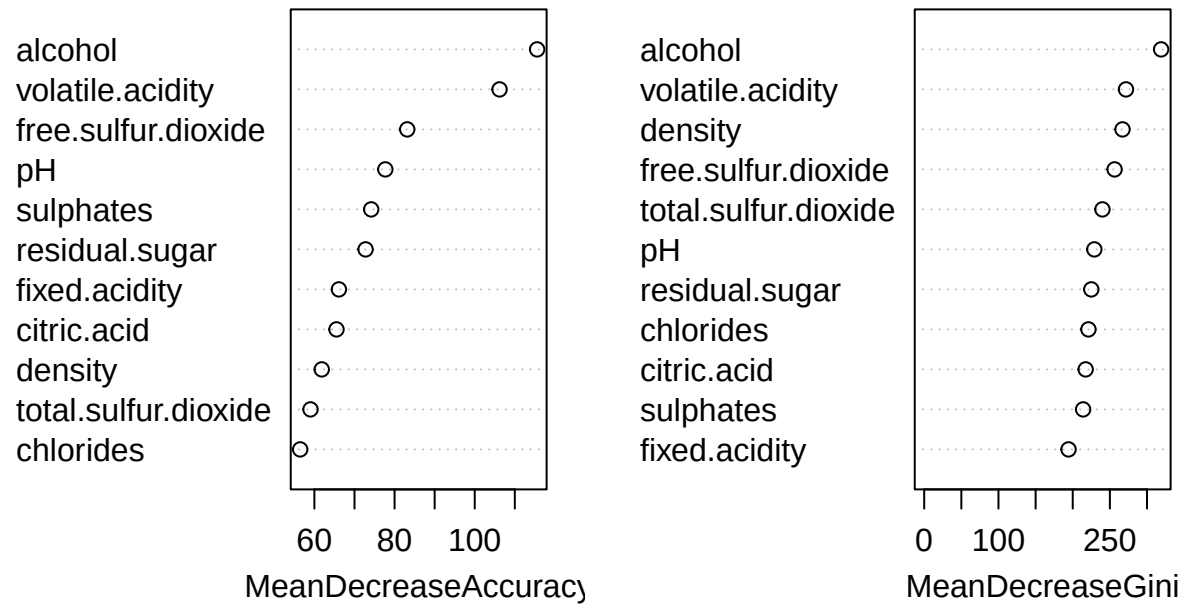
```
##
```

```
## Call:
##  randomForest(formula = quality ~ ., data = dfw.train, mtry = 5,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 30.95%
## Confusion matrix:
##   3  4   5    6   7  8 9 class.error
## 3 0  0   6   10   0  0 0   1.0000000
## 4 0 37  61   44   1  0 0   0.7412587
## 5 0  9 788  332  10  1 0   0.3087719
## 6 0  3 215 1434 108  2 0   0.1861521
## 7 0  0  12  313 384  4 0   0.4614306
## 8 0  0   3   42  32 63 0   0.5500000
## 9 0  0   0    3   2  0 0   1.0000000
```

```
importance(rf.dfw)
```

```
##                                3        4         5        6        7        8
## fixed.acidity         0.934691171 16.79631  41.93579 40.82768 37.77254 32.42080
## volatile.acidity      0.022982615 36.65051  75.95380 67.18756 71.51238 56.49942
## citric.acid           1.325451397 27.79544  45.00530 45.79115 42.63261 35.07976
## residual.sugar       -0.338191277 16.85781  47.44687 50.07990 40.00351 32.43419
## chlorides             3.253556390 22.31596  42.03682 34.02901 48.45844 31.85674
## free.sulfur.dioxide   3.463226071 35.10500  49.33115 54.71735 47.77514 39.17291
## total.sulfur.dioxide  2.286121431 19.97875  38.35331 37.02814 47.17408 33.41861
## density              -0.997868485 17.07368  36.62279 45.92075 45.31221 37.18234
## pH                   -0.005083823 15.70966  50.51408 43.29270 52.75925 34.40339
## sulphates            -0.923087982 17.05810  44.03648 47.77855 43.40788 41.93533
## alcohol              -1.938071492 26.26440 101.51608 51.60135 77.12848 68.49893
##                               9 MeanDecreaseAccuracy MeanDecreaseGini
## fixed.acidity        -0.4041270             66.11081         194.3583
## volatile.acidity      1.4170505            106.19627         271.7197
## citric.acid           1.4128599             65.51583         217.3960
## residual.sugar        2.2779871             72.77112         224.9990
## chlorides             1.1948196             56.45945         221.1430
## free.sulfur.dioxide  -1.9946879             83.16098         256.3581
## total.sulfur.dioxide  0.5424857             59.02295         239.9798
## density               1.8445339             61.83270         267.0266
## pH                    0.6549344             77.69129         229.0779
## sulphates             0.2773714             74.16739         213.9687
## alcohol               0.3991070            115.56474         319.0162
```

```
varImpPlot(rf.dfw)
```




```
#Training-test shows a classification of ~65-66% for random forest

dfw.train2 <- dplyr::select(dfw.train, quality, alcohol, volatile.acidity,
                            density,free.sulfur.dioxide)
dfw.test2 <- dplyr::select(dfw.test, quality, alcohol, volatile.acidity,
                           density,free.sulfur.dioxide)

rf.dfw2 <- randomForest(quality~., data=dfw.train2, mtry=2, importance=TRUE)
rf.dfw.pred2 <- predict(rf.dfw2, newdata = dfw.test2)

mean(rf.dfw.pred2==dfw.test2$quality)
```

```
## [1] 0.6578141
```

```
rf.dfw2
```

```
##
## Call:
##  randomForest(formula = quality ~ ., data = dfw.train2, mtry = 2,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 33.99%
## Confusion matrix:
```

```
##   3  4   5    6   7  8 9 class.error
## 3 0  1   7    8   0  0 0   1.0000000
## 4 0 37  63   40   3  0 0   0.7412587
## 5 1 17 782  318  20  2 0   0.3140351
## 6 0  5 251 1322 176  6 2   0.2497162
## 7 0  0  23  295 383 12 0   0.4628331
## 8 0  1   2   42  32 63 0   0.5500000
## 9 0  0   1    2   1  1 0   1.0000000
```

```
#Four best from MDI, gives a classification rate of ~64.65%

dfw.train3 <- dplyr::select(dfw.train, quality, alcohol, volatile.acidity,
                            pH,free.sulfur.dioxide)
dfw.test3 <- dplyr::select(dfw.test, quality, alcohol, volatile.acidity,
                           pH,free.sulfur.dioxide)

rf.dfw3 <- randomForest(quality~., data=dfw.train3, mtry=2, importance=TRUE)
rf.dfw.pred3 <- predict(rf.dfw3, newdata = dfw.test3)

mean(rf.dfw.pred3==dfw.test3$quality)
```

```
## [1] 0.6670072
```

```
rf.dfw3
```

```
##
## Call:
##  randomForest(formula = quality ~ ., data = dfw.train3, mtry = 2,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 33.86%
## Confusion matrix:
##   3  4   5    6   7  8 9 class.error
## 3 0  0   5   11   0  0 0   1.0000000
## 4 0 42  56   43   2  0 0   0.7062937
## 5 0 12 768  333  26  1 0   0.3263158
## 6 1  8 258 1329 154 12 0   0.2457435
## 7 0  1  18  291 392 11 0   0.4502104
## 8 0  1   2   35  41 61 0   0.5642857
## 9 0  0   0    4   1  0 0   1.0000000
```

```
#Four best from MDA, gives a classifiation rate of ~62.9
```