White_CART_Final.R

nebojsahrnjez

2021-12-01

```
library(readr)
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
```

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

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(moments)
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(ggplot2)
library(ggrepel)
library(gridExtra)
```

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

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

```
#Libraries from exploratory analysis

library(cvTools)
```

```
## Loading required package: lattice

## Loading required package: robustbase
```

```
library(rpart)
library(rpart.plot)
library(rpartScore)
#Libraries for this script

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

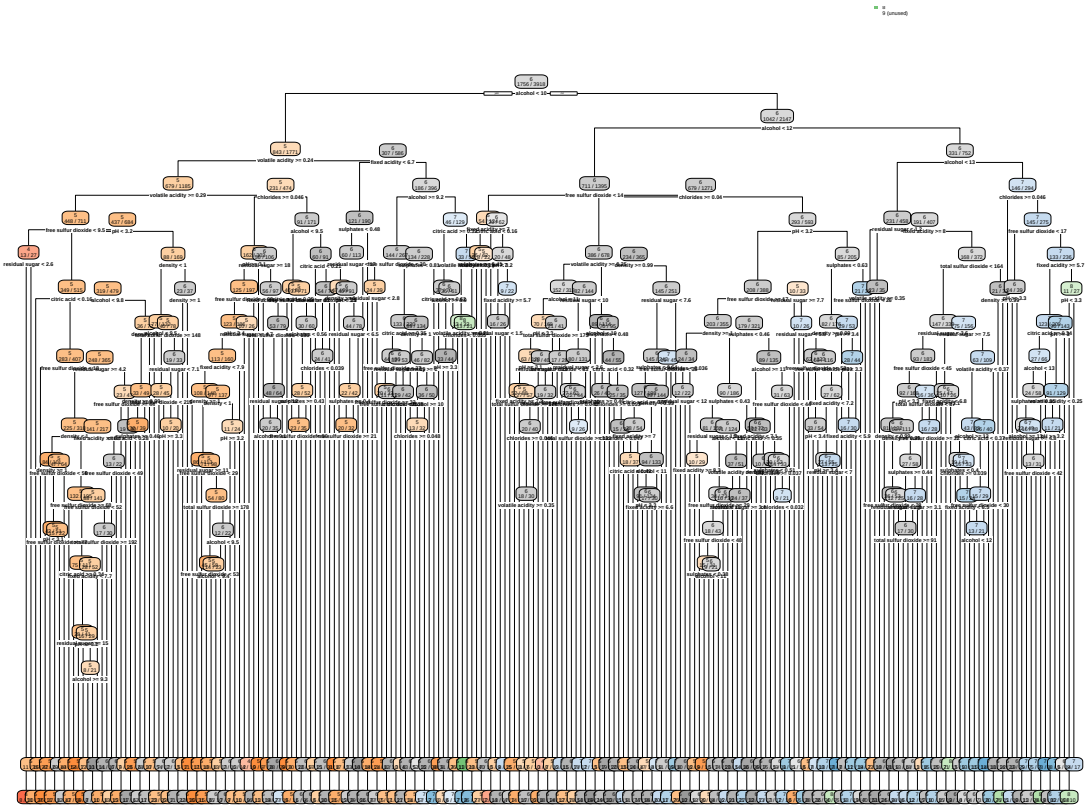
```
## Rows: 4898 Columns: 12

## -- 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))
```

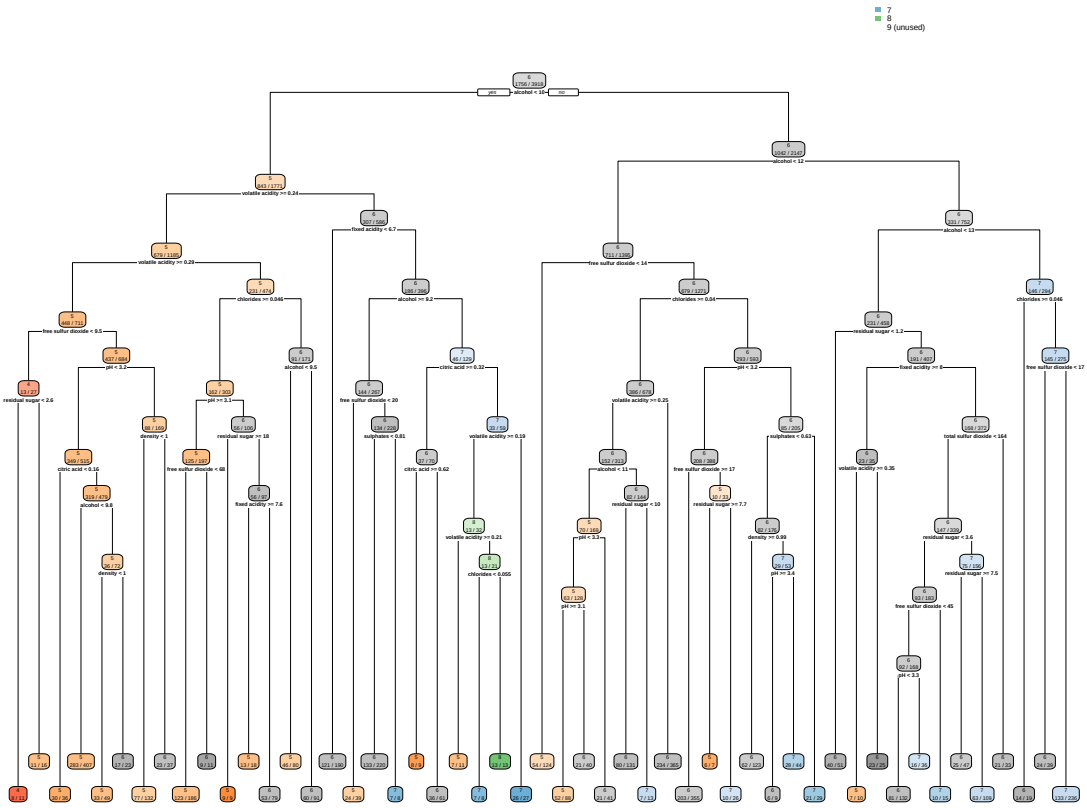
```
## [1] 0
```

```r
white <- na.omit(white)
#Reading in the data

dfw <- as.data.frame(white)

dfw <- dfw[-2782,]
dfw2 <- subset(dfw, select = -density)

#Creating the dataframes to be used

CART.results <- data.frame(matrix(ncol=2,nrow=0,
              dimnames=list(NULL, c("Model", "Classification Accuracy %"))))

#Empty data frame for results

set.seed(100)

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

#Training and Test sets for dfw

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

w.tree <- rpart(quality~., data = dfw.train, method = "class", cp = 0.000001)

w.tree.pred <- predict(w.tree, newdata = dfw.test, type = "class")

w.tree.class <- mean(w.tree.pred == dfw.test$quality)

CART.results[1,] <- c("dfw CART w/ Train/Test", round(w.tree.class*100,
                                                      digits = 2))

#dfw simple Classifcation Tree

rpart.plot(w.tree, extra = 2, digits = 2)
```

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

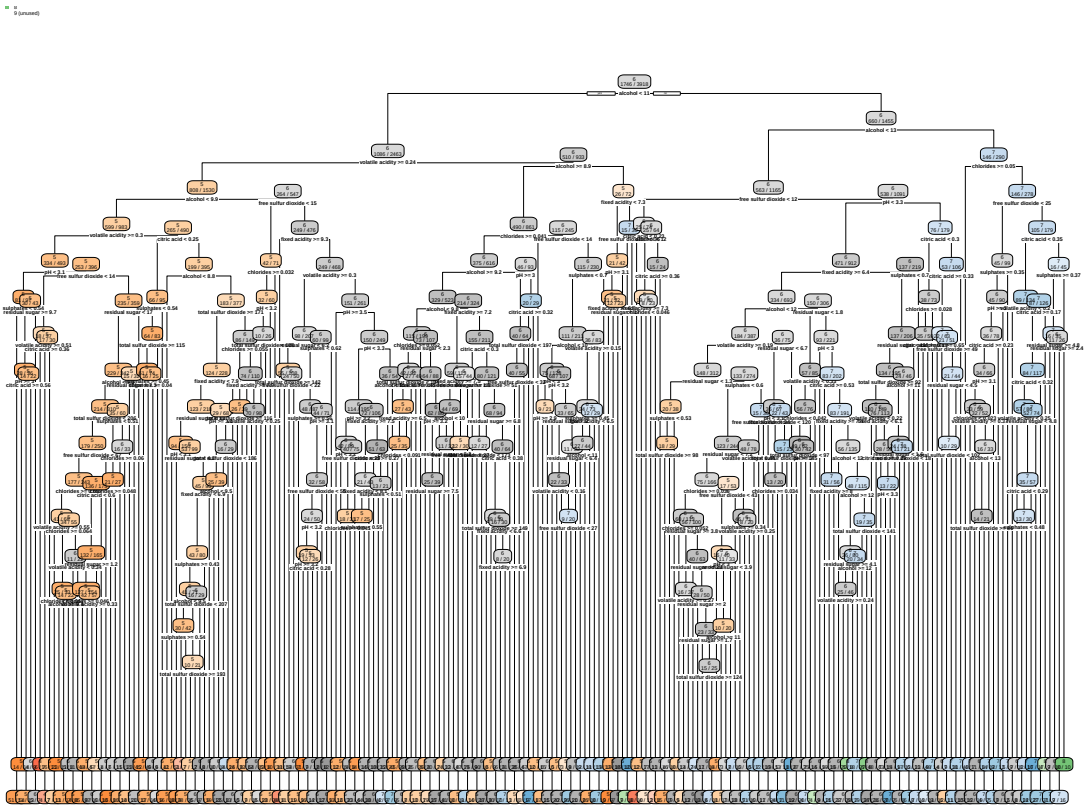


```
#plot dfw simple classification tree

cp <- data.frame(w.tree$cptable)
min.cp <- which.min(cp$xerror)
cp <- cp$CP[min.cp]

#Find cp with minimum relative error for dfw classification tree

w.tree.prun <- prune(w.tree, cp = cp)

w.tree.pred <- predict(w.tree.prun, newdata = dfw.test, type = "class")

w.tree.class <- mean(w.tree.pred == dfw.test$quality)

CART.results[2,] <- c("dfw pruned CART w/ Train/Test", round(w.tree.class*100,
                                                    digits = 2))

#Created pruned min-error tree for dfw with train/test set

rpart.plot(w.tree.prun, extra = 2, digits = 2)
```



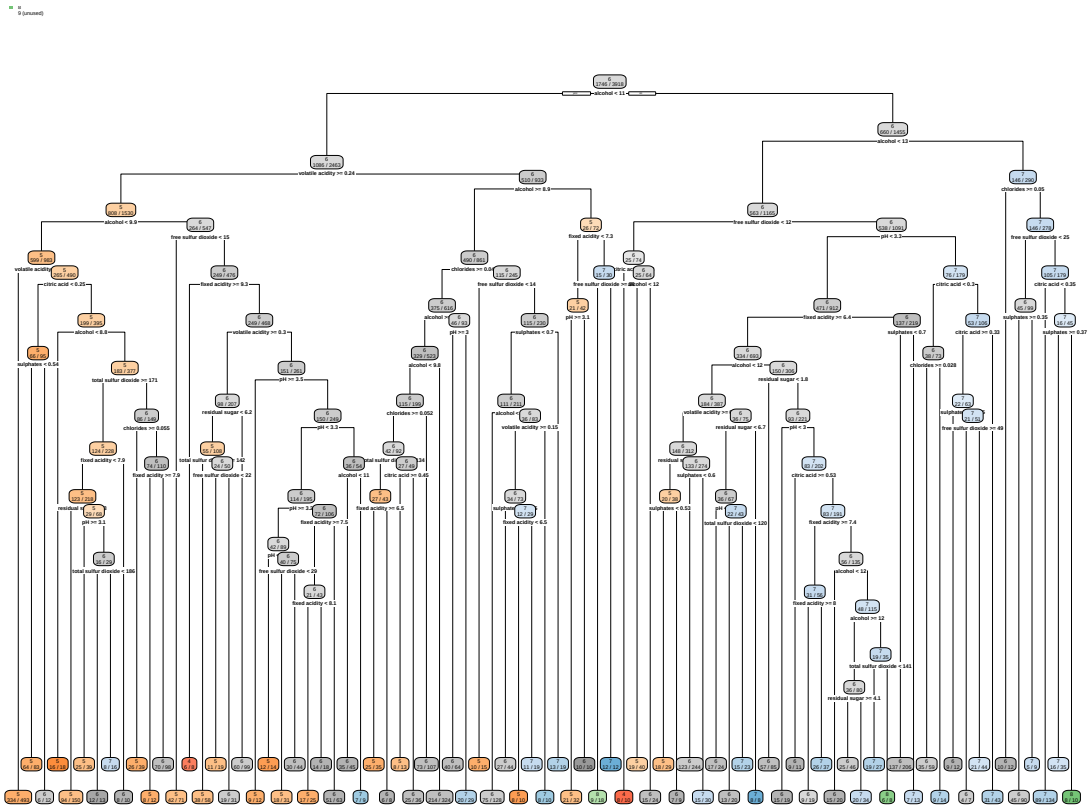
```
#Plot pruned dfw classification tree

test <- sample(1:nrow(dfw2), size = nrow(dfw2)/5)
train <- (-test)

dfw.train2 <- dfw2[train,]
dfw.test2 <- dfw2[test,]

#training and testset for dfw2

w.tree2 <- rpart(quality~., data = dfw.train2, method = "class", cp = 0.000001)

w.tree.pred2 <- predict(w.tree2, newdata = dfw.test2, type = "class")

w.tree.class2 <- mean(w.tree.pred2 == dfw.test2$quality)

CART.results[3,] <- c("dfw2 CART w/ Train/Test", round(w.tree.class2*100,
                                                       digits = 2))
#dfw2 simple Classifcation Tree

rpart.plot(w.tree2, extra = 2, digits = 2)
```

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



```
#plot dfw2 simple classification tree

cp2 <- data.frame(w.tree2$cptable)
min.cp2 <- which.min(cp2$xerror)
cp2 <- cp2$CP[min.cp2]

#Find cp with minimum relative error for dfw2 classification tree

w.tree.prun2 <- prune(w.tree2, cp = cp2)

w.tree.pred2 <- predict(w.tree.prun2, newdata = dfw.test2, type = "class")

w.tree.class2 <- mean(w.tree.pred2 == dfw.test2$quality)

CART.results[4,] <- c("dfw2 pruned CART w/ Train/Test", round(w.tree.class2*100,
                                                              digits = 2))

#Created pruned min-error tree for dfw with train/test set

rpart.plot(w.tree.prun2, extra = 2, digits = 2)
```

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



```
#Plot pruned dfw2 classification tree

k <- 10 #number of folds

folds <- cvFolds(nrow(dfw), K=k)
folds2 <- cvFolds(nrow(dfw2), K=k)

w.tree.cv.class <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))
w.tree.cv.class2 <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))

#Preparing both datasets for cross-validation

for(i in 1:k){
  tr.tree <- dfw[folds$subsets[folds$which != i],]
  te.tree <- dfw[folds$subsets[folds$which == i],]

  w.tree.cv <- rpart(quality~., data = tr.tree, method = "class",
                     cp = 0.000001)

  cp.cv <- data.frame(w.tree.cv$cptable)
  min.cp.cv <- which.min(cp.cv$xerror)
  cp.cv <- cp.cv$CP[min.cp.cv]

  w.tree.prun.cv <- prune(w.tree.cv, cp = cp.cv)

  w.tree.pred.cv <- predict(w.tree.prun.cv, newdata = te.tree, type = "class")
```

```
  w.tree.cv.class[i] <- mean(w.tree.pred.cv == te.tree$quality)

}

w.tree.cv.class
```

```
##               1
##  [1,] 0.5734694
##  [2,] 0.5775510
##  [3,] 0.5163265
##  [4,] 0.5489796
##  [5,] 0.5571429
##  [6,] 0.5163265
##  [7,] 0.5469388
##  [8,] 0.5807771
##  [9,] 0.5480573
## [10,] 0.5030675
```

```
w.tree.cv.class <- mean(w.tree.cv.class)
print(paste("The average outputs correctly predicted is",
            round(w.tree.cv.class*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 54.69 %"
```

```
CART.results[5,] <- c("dfw CART w/ 10-fold CV", round(w.tree.cv.class*100,
                                                      digits=2))

#dfw pruned classification tree w/ cross-validation

for(i in 1:k){
  tr.tree2 <- dfw2[folds2$subsets[folds2$which != i],]
  te.tree2 <- dfw2[folds2$subsets[folds2$which == i],]

  w.tree.cv2 <- rpart(quality~., data = tr.tree2, method = "class",
                      cp = 0.000001)

  cp.cv2 <- data.frame(w.tree.cv2$cptable)
  min.cp.cv2 <- which.min(cp.cv2$xerror)
  cp.cv2 <- cp.cv2$CP[min.cp.cv2]

  w.tree.prun.cv2 <- prune(w.tree.cv2, cp = cp.cv2)

  w.tree.pred.cv2 <- predict(w.tree.prun.cv2, newdata = te.tree2,
                             type = "class")

  w.tree.cv.class2[i] <- mean(w.tree.pred.cv2 == te.tree2$quality)

}

w.tree.cv.class2
```

```
##               1
```

```
##  [1,] 0.5408163
##  [2,] 0.5387755
##  [3,] 0.5673469
##  [4,] 0.5428571
##  [5,] 0.5979592
##  [6,] 0.5285714
##  [7,] 0.5326531
##  [8,] 0.5357873
##  [9,] 0.5378323
## [10,] 0.6421268
```

```
w.tree.cv.class2 <- mean(w.tree.cv.class2)
print(paste("The average outputs correctly predicted is",
            round(w.tree.cv.class2*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 55.65 %"
```

```
CART.results[6,] <- c("dfw2 CART w/ 10-fold CV", round(w.tree.cv.class2*100,
                                                       digits=2))

#dfw2 pruned classification tree w/ cross-validation


w.ordtree <- rpartScore(quality~., data = dfw.train,prune = "mr",cp= 0.000001)

w.ordtree.pred <- predict(w.ordtree, newdata = dfw.test)

w.ordtree.class <- mean(w.ordtree.pred == dfw.test$quality)

CART.results[7,] <- c("dfw Ordinal Tree w/ Train/Test",
                      round(w.ordtree.class*100,digits = 2))

#dfw simple ordinal tree with training/test set

ordcp <- data.frame(w.ordtree$cptable)
min.ordcp <- which.min(ordcp$xerror)
ordcp <- ordcp$CP[min.ordcp]

#Find cp with minimum relative error for dfw ordinal tree

w.ordtree.prun <- prune(w.ordtree, cp = ordcp)

w.ordtree.pred <- predict(w.ordtree.prun, newdata = dfw.test)

w.ordtree.class <- mean(w.ordtree.pred == dfw.test$quality)

CART.results[8,] <- c("dfw pruned Ordinal Tree w/ Train/Test",
                      round(w.ordtree.class*100,digits = 2))

#dfw pruned ordinal tree with training/test set

w.ordtree2 <- rpartScore(quality~., data = dfw.train2,prune = "mr",cp= 0.000001)
```

```
w.ordtree.pred2 <- predict(w.ordtree2, newdata = dfw.test2)

w.ordtree.class2 <- mean(w.ordtree.pred2 == dfw.test2$quality)

CART.results[9,] <- c("dfw2 Ordinal Tree w/ Train/Test",
                      round(w.ordtree.class2*100,digits = 2))

#dfw2 simple ordinal tree with training/test set

ordcp2 <- data.frame(w.ordtree2$cptable)
min.ordcp2 <- which.min(ordcp2$xerror)
ordcp2 <- ordcp2$CP[min.ordcp2]

#Find cp with minimum relative error for dfw2 ordinal tree

w.ordtree.prun2 <- prune(w.ordtree2, cp = ordcp2)

w.ordtree.pred2 <- predict(w.ordtree.prun2, newdata = dfw.test2)

w.ordtree.class2 <- mean(w.ordtree.pred2 == dfw.test2$quality)

CART.results[10,] <- c("dfw2 pruned Ordinal Tree w/ Train/Test",
                      round(w.ordtree.class2*100,digits = 2))


#dfw2 pruned ordinal tree with training/test set

w.ordtree.class.cv <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))
w.ordtree.class.cv2 <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))

#preparing for cross-validation

for(i in 1:k){
  tr.ord <- dfw[folds$subsets[folds$which != i],]
  te.ord <- dfw[folds$subsets[folds$which == i],]

  w.ordtree.cv <- rpartScore(quality~., data = tr.ord,prune = "mr",
                             cp=0.000001)

  ordcp.cv <- data.frame(w.ordtree.cv$cptable)
  min.ordcp.cv <- which.min(ordcp.cv$xerror)
  ordcp.cv <- ordcp.cv$CP[min.ordcp.cv]

  w.ordtree.prun.cv <- prune(w.ordtree.cv, cp = ordcp.cv)

  w.ordtree.pred.cv <- predict(w.ordtree.prun.cv, newdata = te.ord)

  w.ordtree.class.cv[i] <- mean(w.ordtree.pred.cv == te.ord$quality)

}

w.ordtree.class.cv
```

```
##               1
##  [1,] 0.5734694
##  [2,] 0.5857143
##  [3,] 0.5387755
##  [4,] 0.5448980
##  [5,] 0.5489796
##  [6,] 0.5285714
##  [7,] 0.5530612
##  [8,] 0.5787321
##  [9,] 0.5541922
## [10,] 0.5235174
```

```
w.ordtree.class.cv <- mean(w.ordtree.class.cv)
print(paste("The average outputs correctly predicted is",
            round(w.ordtree.class.cv*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 55.3 %"
```

```
CART.results[11,] <- c("dfw pruned ordinal tree w/ 10-fold CV",
                       round(w.ordtree.class.cv*100,digits=2))

#dfw pruned ordinal tree with cross-validation

for(i in 1:k){
  tr.ord <- dfw2[folds2$subsets[folds2$which != i],]
  te.ord <- dfw2[folds2$subsets[folds2$which == i],]

  w.ordtree.cv2 <- rpartScore(quality~., data = tr.ord,prune = "mr",
                              cp=0.000001)

  ordcp.cv2 <- data.frame(w.ordtree.cv2$cptable)
  min.ordcp.cv2 <- which.min(ordcp.cv2$xerror)
  ordcp.cv2 <- ordcp.cv2$CP[min.ordcp.cv2]

  w.ordtree.prun.cv2 <- prune(w.ordtree.cv2, cp = ordcp.cv2)

  w.ordtree.pred.cv2 <- predict(w.ordtree.prun.cv2, newdata = te.ord)

  w.ordtree.class.cv2[i] <- mean(w.ordtree.pred.cv2 == te.ord$quality)

}

w.ordtree.class.cv2
```

```
##               1
##  [1,] 0.5265306
##  [2,] 0.5959184
##  [3,] 0.5877551
##  [4,] 0.5673469
##  [5,] 0.5408163
##  [6,] 0.5469388
##  [7,] 0.5346939
##  [8,] 0.5214724
```

```
##  [9,] 0.5460123
## [10,] 0.5807771
```

```
w.ordtree.class.cv2 <- mean(w.ordtree.class.cv2)
print(paste("The average outputs correctly predicted is",
            round(w.ordtree.class.cv2*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 55.48 %"
```

```
CART.results[12,] <- c("dfw2 pruned ordinal tree w/ 10-fold CV",
                       round(w.ordtree.class.cv2*100,digits=2))

#dfw2 pruned ordinal tree with cross-validation

CART.results
```

```
##                                    Model Classification.Accuracy..
## 1                  dfw CART w/ Train/Test                     55.77
## 2            dfw pruned CART w/ Train/Test                     55.46
## 3                 dfw2 CART w/ Train/Test                     54.44
## 4           dfw2 pruned CART w/ Train/Test                     56.49
## 5                  dfw CART w/ 10-fold CV                     54.69
## 6                 dfw2 CART w/ 10-fold CV                     55.65
## 7           dfw Ordinal Tree w/ Train/Test                     55.77
## 8    dfw pruned Ordinal Tree w/ Train/Test                     55.36
## 9          dfw2 Ordinal Tree w/ Train/Test                     56.89
## 10 dfw2 pruned Ordinal Tree w/ Train/Test                     58.63
## 11  dfw pruned ordinal tree w/ 10-fold CV                      55.3
## 12 dfw2 pruned ordinal tree w/ 10-fold CV                     55.48
```