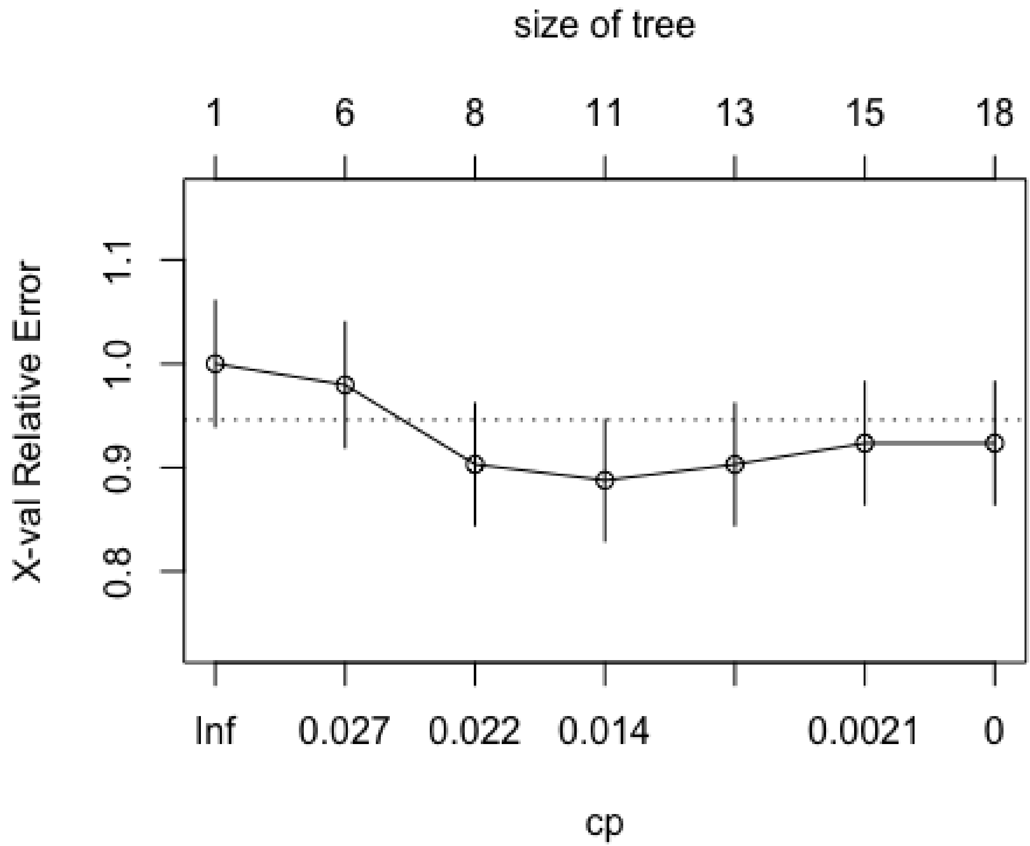
assignment4-part2

yunzhi wang

```
set.seed(600)
library(caret)

## Warning: package 'caret' was built under R version 3.4.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone
'zone/tz/2018c.
## 1.0/zoneinfo/America/Chicago'

library(rpart)
data("GermanCredit")
mydata <- GermanCredit
#names(mydata)
mydata.split <- sample(1:nrow(mydata), size = 0.7 * nrow(mydata))
Train <-
mydata[mydata.split,c(10,1,2,3,9,11,12,13,15,16,17,20,22,23,25,26,29,31,32,38
,43,46,47,53,57,59)]
Holdout <- mydata[-mydata.split,
c(10,1,2,3,9,11,12,13,15,16,17,20,22,23,25,26,29,31,32,38,43,46,47,53,57,59)]
```

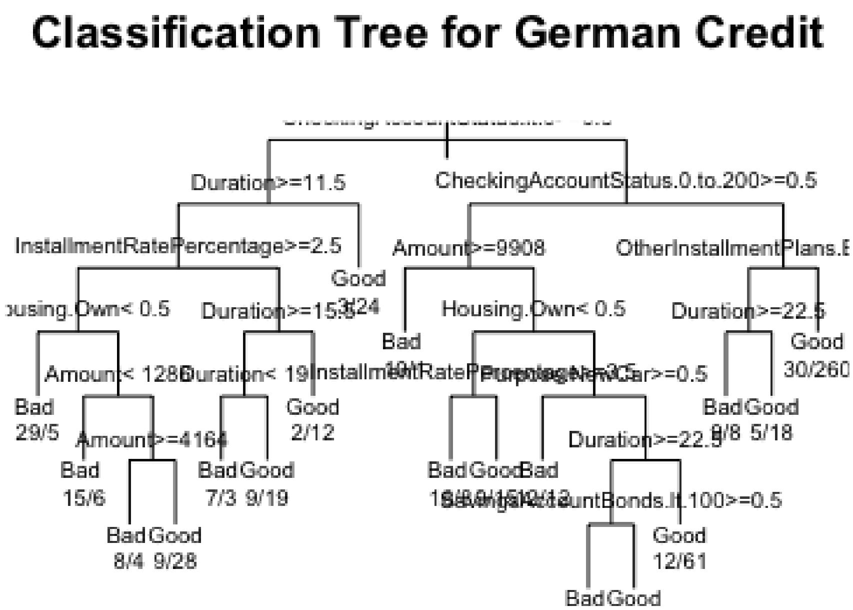
1. build a tree model in which cp = 0, minsplit =30, xval = 10

```
set.seed(600)
tree.min30 <- rpart(formula = Class ~., data = Train,
                    control = rpart.control(cp = 0, minsplit = 30, xval= 10))

set.seed(600)
plotcp(tree.min30)
```



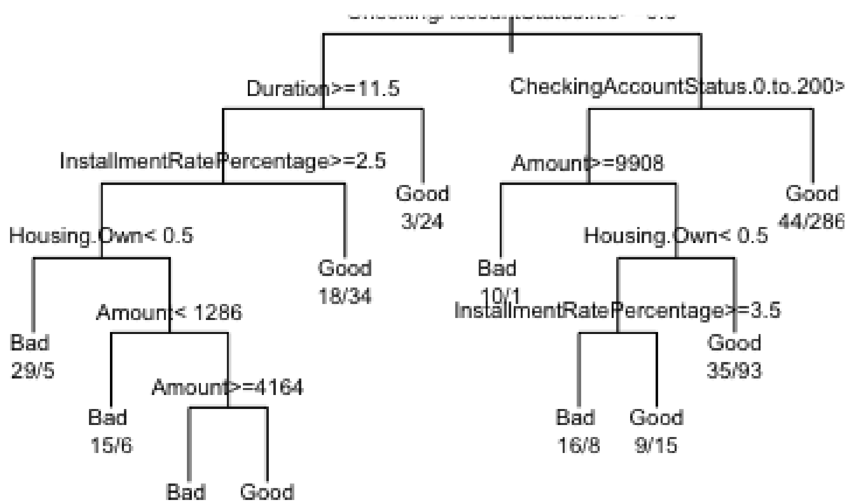

```
printcp(tree.min30)

##
## Classification tree:
## rpart(formula = Class ~ ., data = Train, control = rpart.control(cp = 0,
##     minsplit = 30, xval = 10))
##
## Variables actually used in tree construction:
## [1] Amount                          CheckingAccountStatus.0.to.200
## [3] CheckingAccountStatus.lt.0      Duration
## [5] Housing.Own                     InstallmentRatePercentage
## [7] OtherInstallmentPlans.Bank      Purpose.NewCar
## [9] SavingsAccountBonds.lt.100
##
## Root node error: 196/700 = 0.28
##
## n= 700
##
##           CP nsplit rel error  xerror     xstd
## 1 0.0306122      0   1.00000 1.00000 0.060609
## 2 0.0229592      5   0.83163 0.97959 0.060225
## 3 0.0204082      7   0.78571 0.90306 0.058672
## 4 0.0102041     10   0.72449 0.88776 0.058339
```

```
## 5 0.0025510     12   0.70408 0.90306 0.058672
## 6 0.0017007     14   0.69898 0.92347 0.059104
## 7 0.0000000     17   0.69388 0.92347 0.059104

plot(tree.min30, main="Classification Tree for German Credit", uniform=TRUE)
text(tree.min30, cex=0.6, use.n=TRUE)
```




```
#From the cross validation of different cp values, cp = 0.006 results in the smallest cv error.
tree.min30.prune <- prune(tree.min30, cp =
tree.min30$cptable[which.min(tree.min30$cptable[,"xerror"]),"CP"])
plot(tree.min30.prune,main="Prune Tree cp: German Credit", uniform=TRUE)
text(tree.min30.prune, cex=0.6, use.n=TRUE)
```

Prune Tree cp: German Credit




```r
tree.min30$cptable[which.min(tree.min30$cptable[,"xerror"]),"CP"]

## [1] 0.01020408
```

Interpreatation: From the cross validation of different cp values, the smallest xerror value is 0.88776 , which suggest that when cp=0.0102041 results in the smallest cv error. Therefore we choose cp = 0.0192308, which results in 10 splits.

3.Generate Confusion matrix for the tree

```r
#predict(tree.min30.prune)
set.seed(600)
table(pred.class = predict(tree.min30.prune,type = "class"),true.class =
Train[, "Class"] )

##            true.class
## pred.class Bad Good
##        Bad   78   24
##        Good 118  480
```

Interpretation: Pruned tree model with cp = 0.01020408 predicts the training set with (78+480)/700=80% accuracy,which is a good result.

```
set.seed(618)
table(pred.class = predict(tree.min30.prune, type = "class", newdata =
Holdout),true.class = Holdout[, "Class"] )
```

```
##            true.class
## pred.class Bad Good
##       Bad   30   22
##       Good  74  174
```

Interpretation: The model predicts holdout dataset with (30+174)/300=68% accuracy. This accuracy is lower than the training dataset, but makes sense as testing error rate generallly being higher than training error.

5. Comparision between this model and logistic model.

Interpretation: For the training part, both models gain similar accuracy. 78% for logistic model and 80% for pruned tree model. For the holdout part, logistic model classifies the classes with 72% accuracy while tree model has only 68%. Both models have similar level of bias, but tree model has higher variance thus it is less robust. Logistic model is better than tree model for this dataset.