데이터사이언스의 이해 의사결정 나무

자료 불러오기

ploan <- read.csv("Personal Loan.csv")

**id, zip code 제외 Personal.Loan 변수는 factor로 변환**

ploan <- ploan[,-c(1,5)]  
ploan$Personal.Loan<- as.factor(ploan$Personal.Loan)  
str(ploan)

## 'data.frame': 2500 obs. of 12 variables:  
## $ Age : int 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience : int 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income : int 49 34 11 100 45 29 72 22 81 180 ...  
## $ Family : int 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : int 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : int 0 0 0 0 0 155 0 0 104 0 ...  
## $ Personal.Loan : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...  
## $ Securities.Account: int 1 1 0 0 0 0 0 0 0 0 ...  
## $ CD.Account : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Online : int 0 0 0 0 0 1 1 0 1 0 ...  
## $ CreditCard : int 0 0 0 0 1 0 0 1 0 0 ...

summary(ploan)

## Age Experience Income Family   
## Min. :23.00 Min. :-2.00 Min. : 8.00 Min. :1.000   
## 1st Qu.:35.00 1st Qu.:10.00 1st Qu.: 39.00 1st Qu.:1.000   
## Median :45.00 Median :20.00 Median : 64.00 Median :2.000   
## Mean :45.35 Mean :20.12 Mean : 74.45 Mean :2.408   
## 3rd Qu.:55.00 3rd Qu.:30.00 3rd Qu.: 99.25 3rd Qu.:4.000   
## Max. :67.00 Max. :42.00 Max. :205.00 Max. :4.000   
## CCAvg Education Mortgage Personal.Loan  
## Min. : 0.000 Min. :1.000 Min. : 0.00 0:2244   
## 1st Qu.: 0.700 1st Qu.:1.000 1st Qu.: 0.00 1: 256   
## Median : 1.500 Median :2.000 Median : 0.00   
## Mean : 1.951 Mean :1.866 Mean : 57.39   
## 3rd Qu.: 2.500 3rd Qu.:3.000 3rd Qu.:104.00   
## Max. :10.000 Max. :3.000 Max. :617.00   
## Securities.Account CD.Account Online CreditCard   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :0.0000 Median :1.0000 Median :0.0000

**훈련자료(training data)와 테스트자료(test data)로 자료 쪼개기**

trn\_idx<- 1:1500 # trn <- training data  
ploan\_train <- ploan[trn\_idx,]  
ploan\_test <- ploan[-trn\_idx,]

CART with Post-Pruning

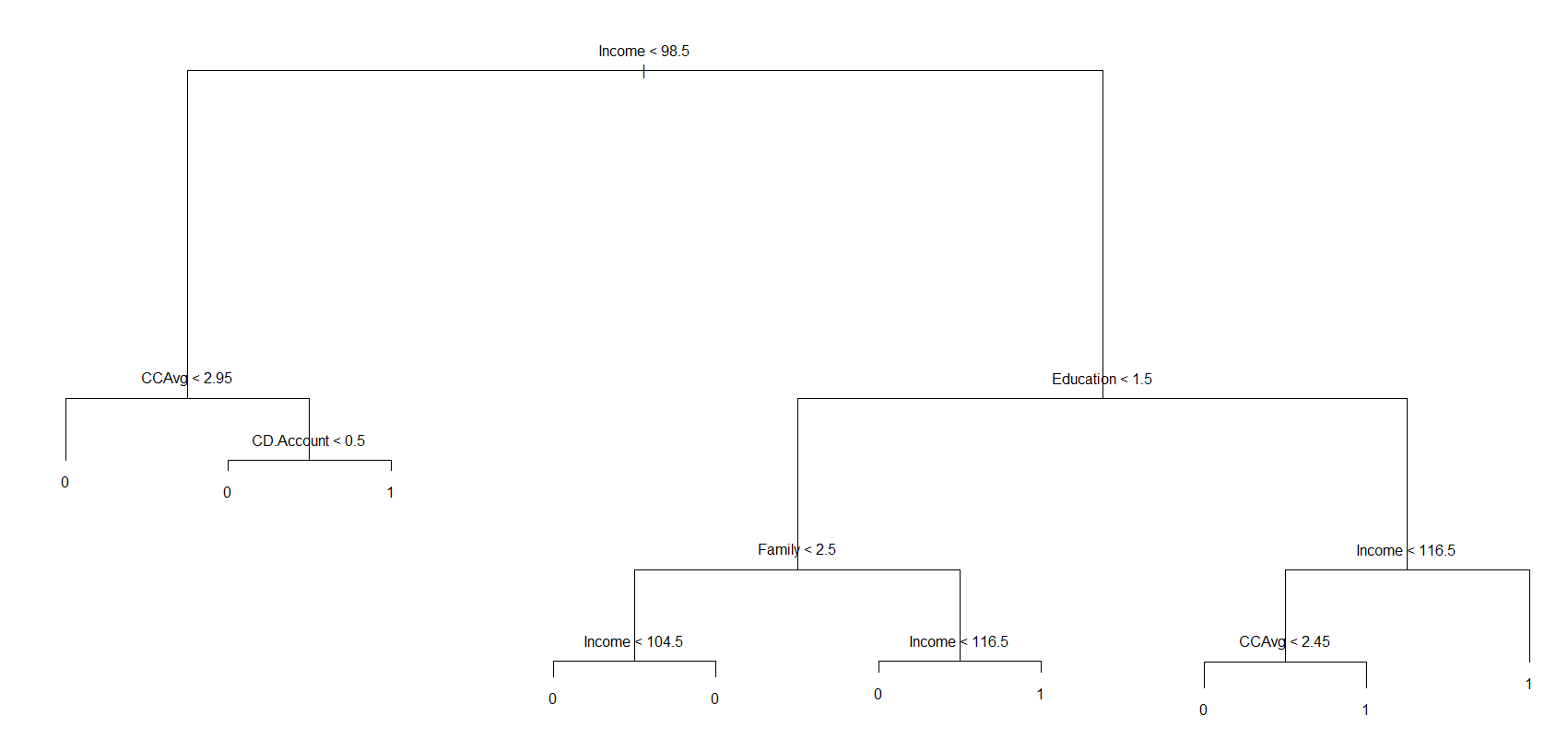
library(tree)

**1단계: Recursive Partitioning(재귀적 분귀)**

# Training the tree  
CART\_post <- tree(Personal.Loan ~ ., ploan\_train)  
summary(CART\_post)

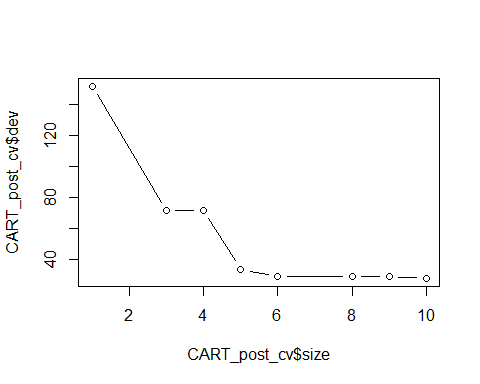
##   
## Classification tree:  
## tree(formula = Personal.Loan ~ ., data = ploan\_train)  
## Variables actually used in tree construction:  
## [1] "Income" "CCAvg" "CD.Account" "Education" "Family"   
## Number of terminal nodes: 10   
## Residual mean deviance: 0.06996 = 104.2 / 1490   
## Misclassification error rate: 0.01267 = 19 / 1500

# Plot the tree  
plot(CART\_post)  
text(CART\_post)



2단계: Post pruning(사후적 가지치기) 및 terminal node 증가에 따른 deviation 값 변화 확인

# Find the best tree  
set.seed(123)  
CART\_post\_cv <- cv.tree(CART\_post, FUN = prune.misclass)  
  
# Plot the pruning result  
plot(CART\_post\_cv$size, CART\_post\_cv$dev, type = "b") #size = # of terminal node

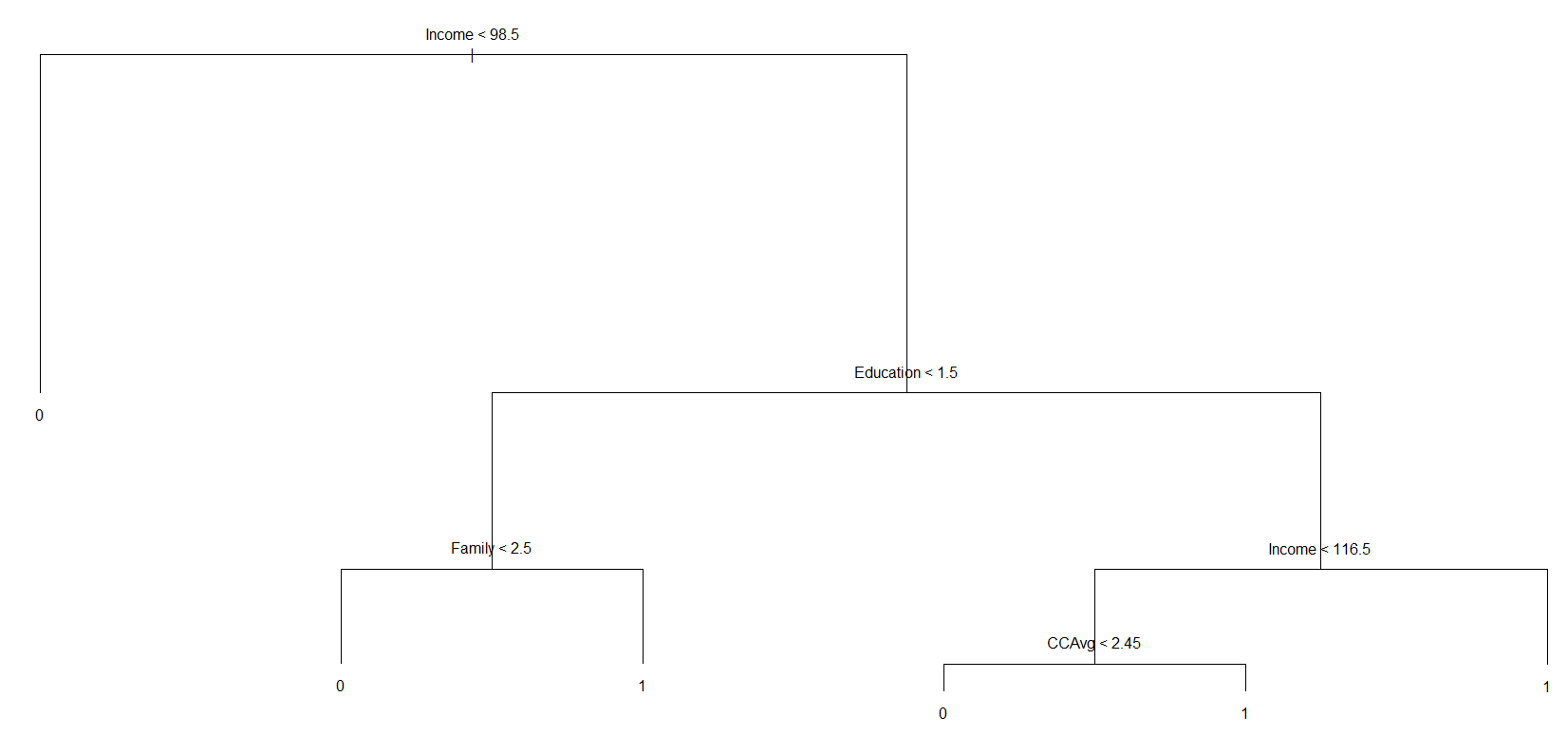


CART\_post\_cv

## $size  
## [1] 10 9 8 6 5 4 3 1  
##   
## $dev  
## [1] 28 29 29 29 34 72 72 152  
##   
## $k  
## [1] -Inf 0.0 1.0 1.5 9.0 17.0 19.0 42.0  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

**3단계: 최종 모형 선택**

# Select the final model  
CART\_post\_pruned <- prune.misclass(CART\_post, best = 6) # best means size of leaf nodes  
plot(CART\_post\_pruned)  
text(CART\_post\_pruned)



test data를 활용하여 모형의 성능 검증

# Prediction  
CART\_post\_prey <- predict(CART\_post\_pruned, ploan\_test, type = "class")  
CART\_post\_cm <- table(ploan\_test$Personal.Loan, CART\_post\_prey)  
CART\_post\_cm

## CART\_post\_prey  
## 0 1  
## 0 888 8  
## 1 11 93

# 사전적 가지치기를 이용한 트리모형 학습

사전적 가지치(post pruning)를 위한 패키지 설치

#install.packages("matrixStats")

#install.packages("party")

library(party)

훈련자료, 검증자료, 테스트자료 쪼개기

# Divide the dataset into training/validation/test datasets  
train\_idx <- 1:1000  
val\_idx <- 1001:1500  
test\_idx <- 1501:2500  
  
ploan\_train <- ploan[train\_idx,]  
ploan\_val <- ploan[val\_idx,]  
ploan\_test <- ploan[test\_idx,]

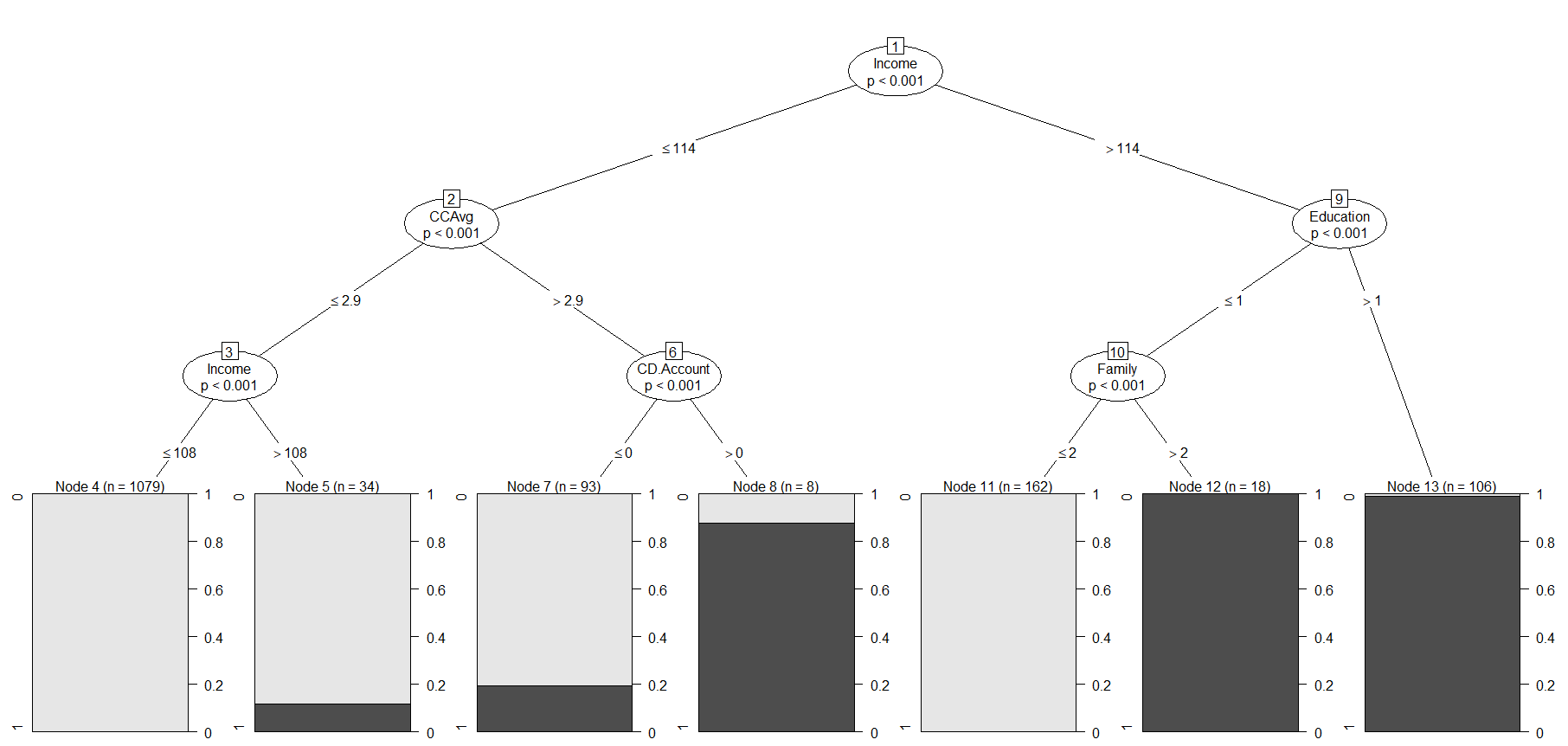
분석하기: pre pruning에서 중요한것은 언제 가지치기를 멈출것이냐 기준 설정 => controls

# Construct single tree and evaluation  
# tree parameter settings  
# min\_criterion = c(0.9, 0.95, 0.99) # 분류 정확도 기준  
# min\_split = c(10, 30, 50, 100) # 마지막 node에서 관측된 최소 관측 값 수, 작게 설정할 수 록 노드가 많아짐 (과적합 가능성 증가)   
# max\_depth = c(0, 10, 5) # 트리의 최대깊이  
  
  
ploan\_control = ctree\_control(mincriterion = 0.90, minsplit = 10, maxdepth = 10)  
ploan\_pretree <- ctree(Personal.Loan ~ ., data = ploan\_train, controls = ploan\_control)

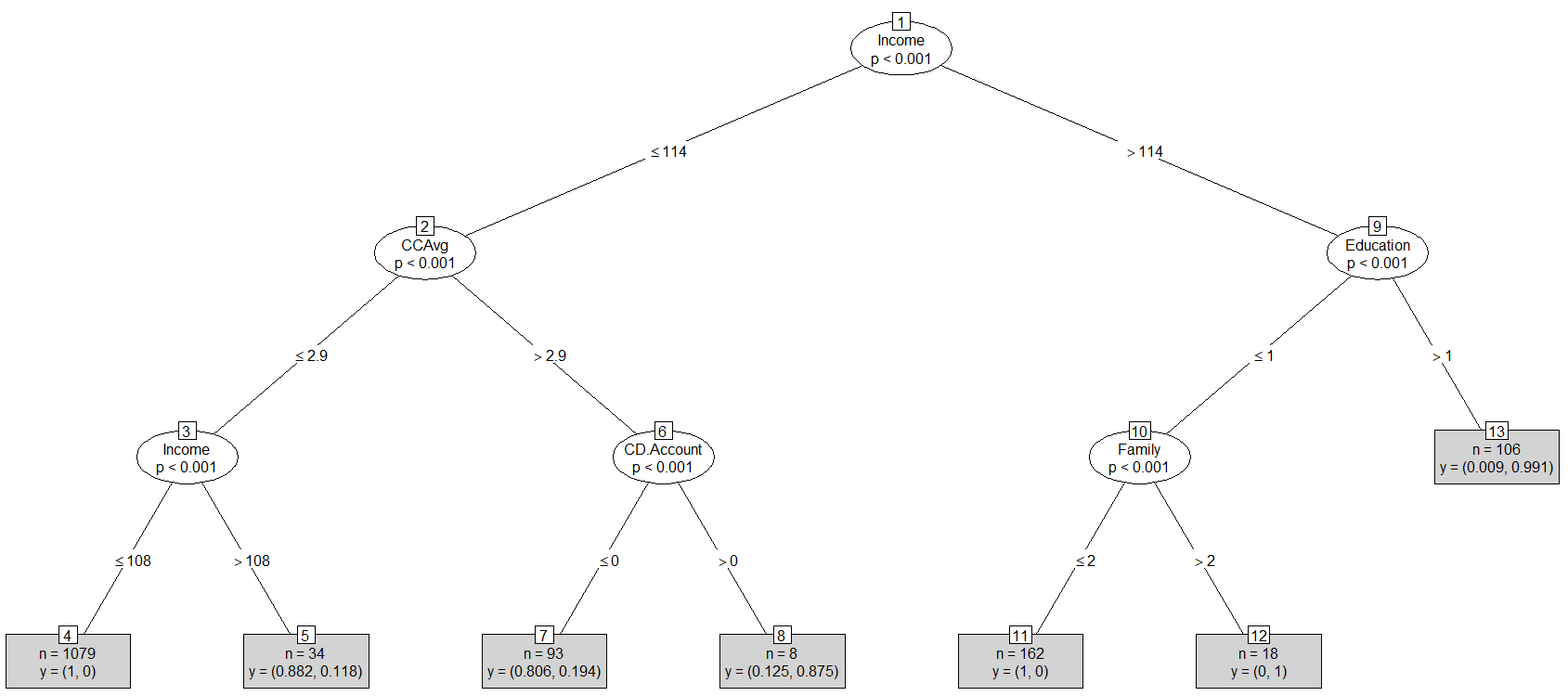
훈련자료와 검증자료를 동시에 활용한 모형 재학습

# Use the training and validation dataset to train the best tree  
ploan\_train <- rbind(ploan\_train, ploan\_val)  
CART\_pre <- ctree(Personal.Loan ~ ., data = ploan\_train, controls = ploan\_control)

# 결과 그림 그리기  
plot(CART\_pre)



plot(CART\_pre, type="simple")



test dat를 활용하여 pre pruning 모형의 성능 검증

CART\_pre\_prediction <- predict(CART\_pre, newdata = ploan\_test)  
  
# Performance of the best tree  
CART\_pre\_cm <- table(ploan\_test$Personal.Loan, CART\_pre\_prediction)  
CART\_pre\_cm

## CART\_pre\_prediction  
## 0 1  
## 0 891 5  
## 1 19 85