**Decision Tree classifier**

Data0<-read.csv("C:/Users/winnie/Documents/R language/DT-Credit.csv",header=TRUE,sep=",")

Data=Data0[,-1]

#Minus the first column

str(Data)

#str():check distinct categories

attach(Data)

#The function attach () can add the data box to the search path of R. After a variable name is encountered, R checks the data box in the search path to locate this variable

library("rpart")

DT\_Model<-rpart(RESPONSE~.,data=Data,control=rpart.control(minsplit = 60,minbucket = 30,maxdepth = 4))

#minsplit：minimum number of observations in a node, minbucket: minimum number of observations in leaf node, Maxdepth: maximum depth for any node. root node’s depth is 0.

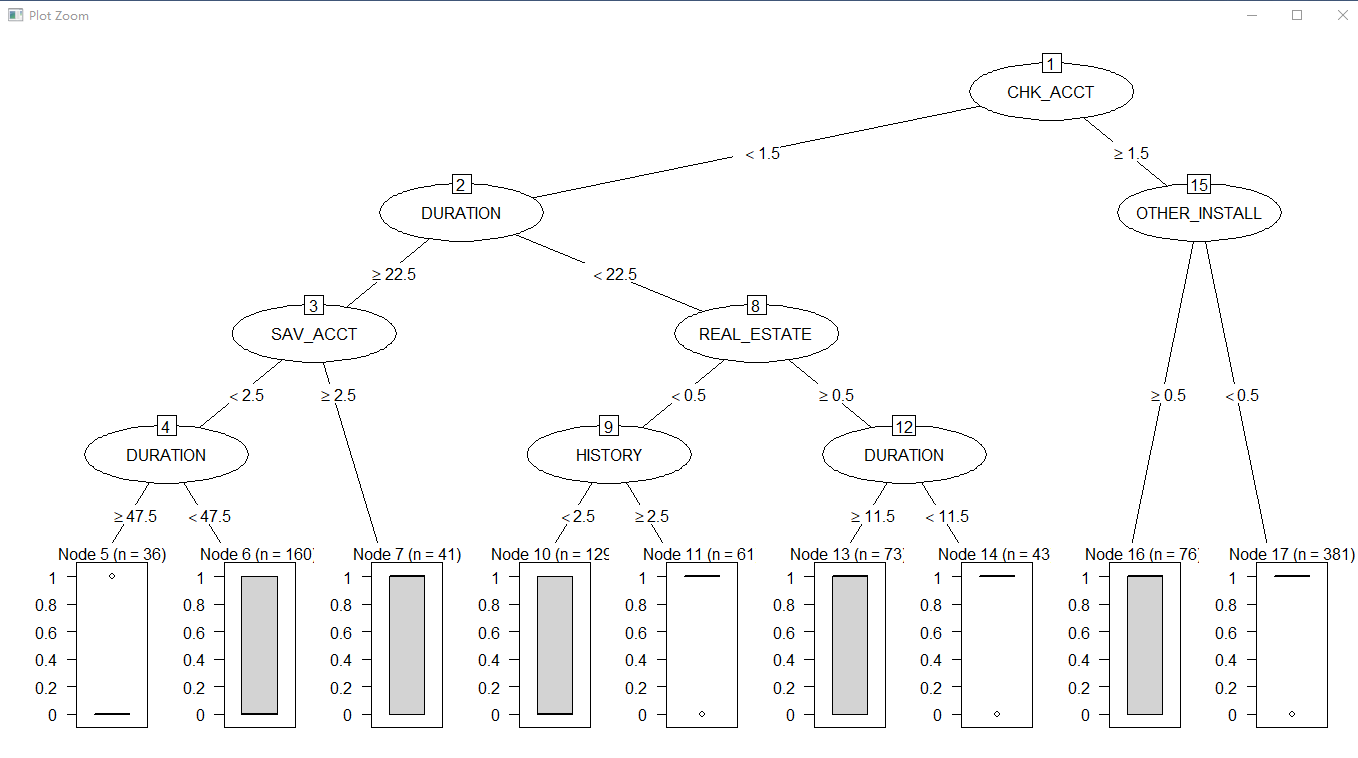
install.packages("partykit")

library("partykit")

plot(as.party(DT\_Model))

print(DT\_Model)

#party is better than rpart as plot function.



**The minimum number of observations in a node for a new spit is 60. The minimum number of observations in leaf node is 30, and I have 5 depth. Because nodes are limited, there are some correct and incorrect categorizations, and the white and grey bars in the leaf node are the percentage of correct and incorrect categorizations**

Target=ifelse(RESPONSE==1,'Y','N')

#ifelse():The replacement function returns a vector.

Data<-data.frame(Data,Target)

#Add Target to main dataset

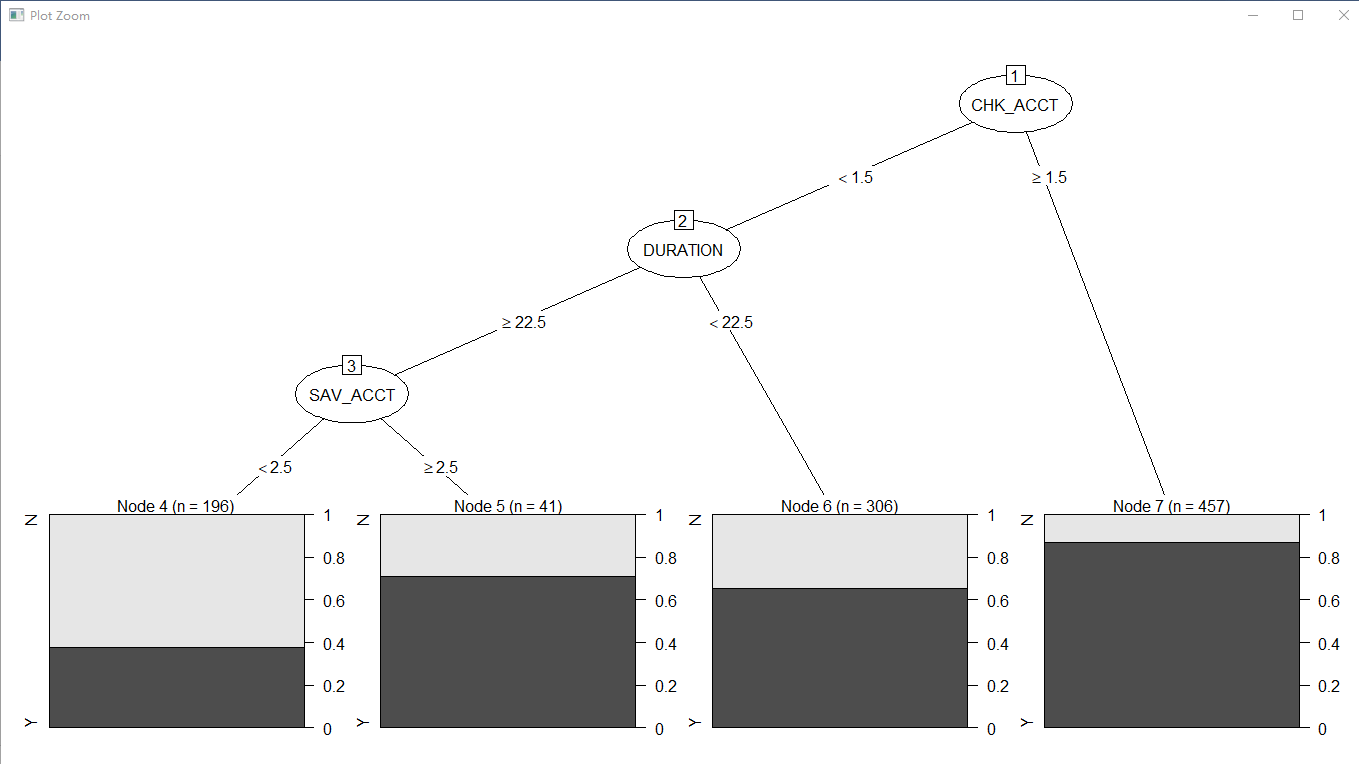
str(Data)

Data1=Data[,-31]

DT\_Model1<-rpart(Target~.,data=Data1,control=rpart.control(minsplit = 60,minbucket = 30,maxdepth = 4))

plot(as.party(DT\_Model1))

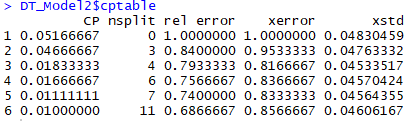
print(DT\_Model1)

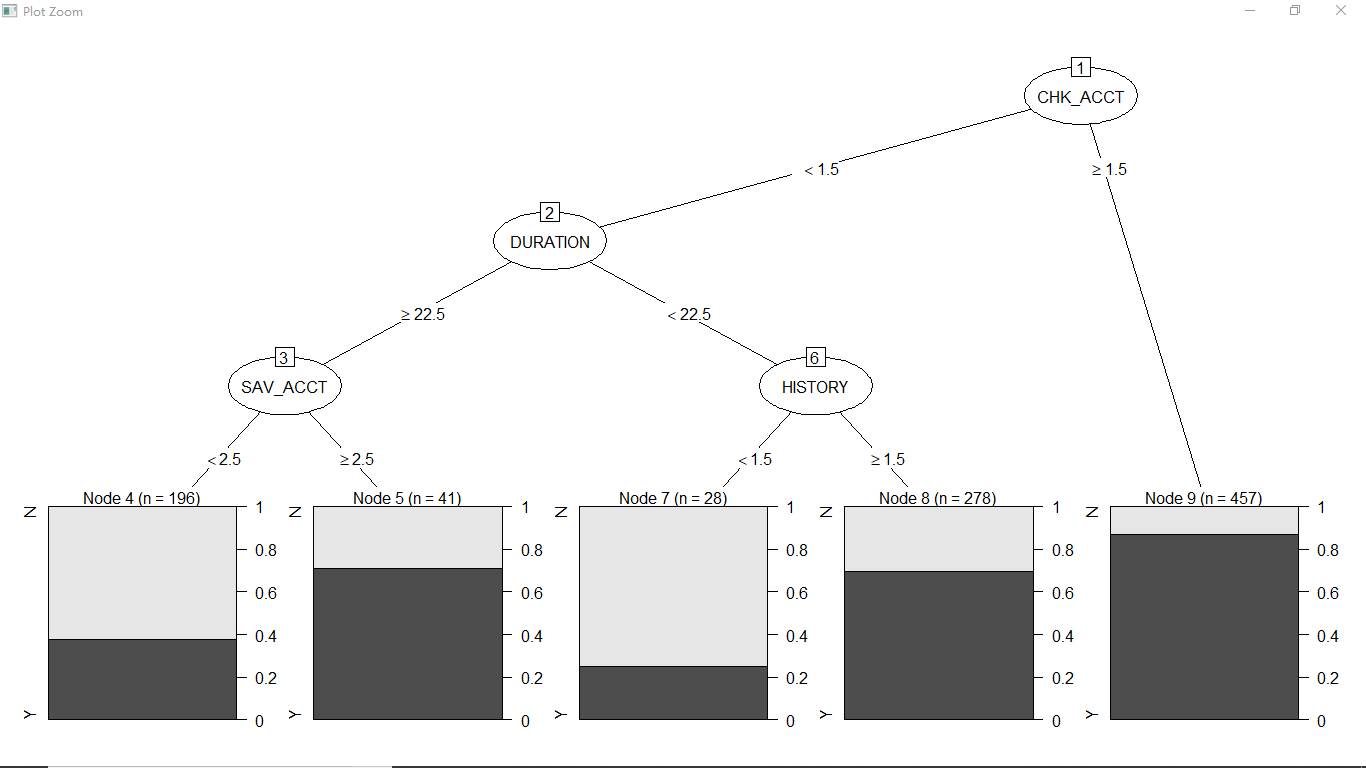


**Actually I don’t know why this result is different from model1, cuz in model2 I just transform 1/0 to Y/N**

DT\_Model2<-rpart(Target~.,data=Data1,control=rpart.control())

#cp：The cp complexity parameter specifies the threshold used for pruning.





**Pre-pruning is to set a threshold control so that when the error before and after splitting exceeds this threshold in the current splitting node, the splitting operation is not performed. The code is ##control=rpart.control(minsplit = 60,minbucket = 30,maxdepth = 4)**

**Post-pruning is to construct a decision tree with training set, and then use the operation of test set to find the minimum variance CP.**

**(1) Pre-threshold setting is very sensitive, a little change, will cause the whole tree very large changes, not easy to set.**

**(2) Pre-pruning is simpler than later pruning. I got 7 nodes in pre-pruning and I got 9 nodes in post-prunning.**

**(3) The results obtained illustrates post-pruning is better.**