Data Analytics Lab

08/10/19

```
rm(list=ls())
```

Step 1 - Import the excel file and checking content of the data

Convert the dataset into csv file, `Read.csv` tells R to read csv file.

```
Data <- read.csv(“DT-Credit.csv”, header=TRUE, sep= “ ; ”)
```

Check distinct categories of Variables using STR function

```
str(Data)
```

you need to change some data as factor:

```
cols <- c(1:2, 4:10, 12:22, 24:32)
```

```
Data[cols] <- lapply(Data[cols], factor)
```

```
str(Data)
```

you need to remove the first column:

```
Data <- Data[,-1]
```

Check your data:

```
names(Data)
```

```
attach(Data)
```

Step 2 - Install package rpart, and click on the checkbox in front of rpart library.

Develop the DT model:

```
DT_Model <-rpart(RESPONSE~., data=Data, control=rpart.control(minsplit=60,
minbucket=30, maxdepth=4 ))
```

minsplit: the minimum number of observations that must exist in a node for a new split
minbucket: the minimum number of observations in any terminal <leaf> node
Maxdepth: Maximum depth for any node, with the root node counted as depth 0.

Step 3 - Install packgae partykit:

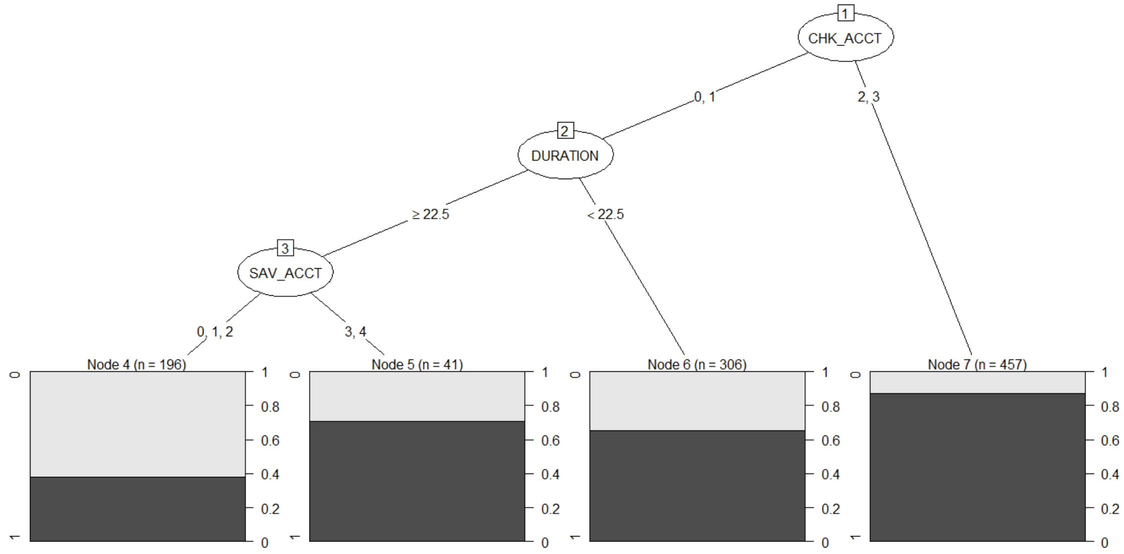
```
install.packages ("partykit")
```

```
library("partykit")
```

```
plot(as.party(DT_Model))
```

```
print(DT_Model)
```

You would get the following output. Describe the results at the end of your scripts.

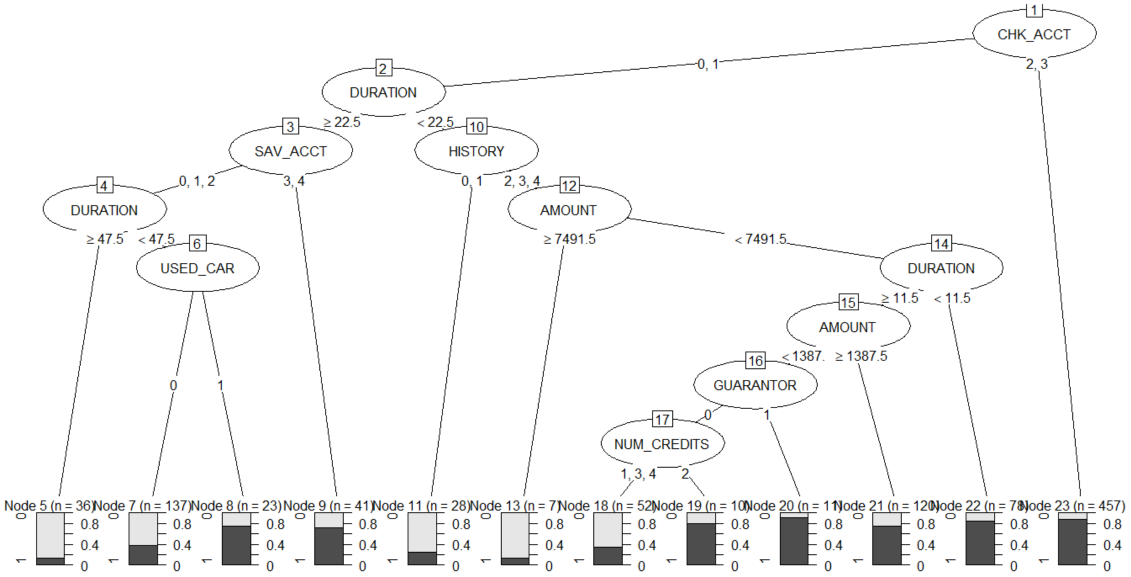


Step 4- Procedure of Pruning

```
DT_Model2<-rpart(Target~., data=Data)
```

```
Plot(as.party(DT_Model2))
```

You should get the following output:



The following line fitted tree's CP table (Matrix of Information on optimal pruning given Complexity Parameter). Look where do you see the least error.

```
print(DT_Model2$cptable)
```

The line below automatically picks up the least error tree

```
opt <- which.min(DT_Model2$cptable [, "xerror"])
```

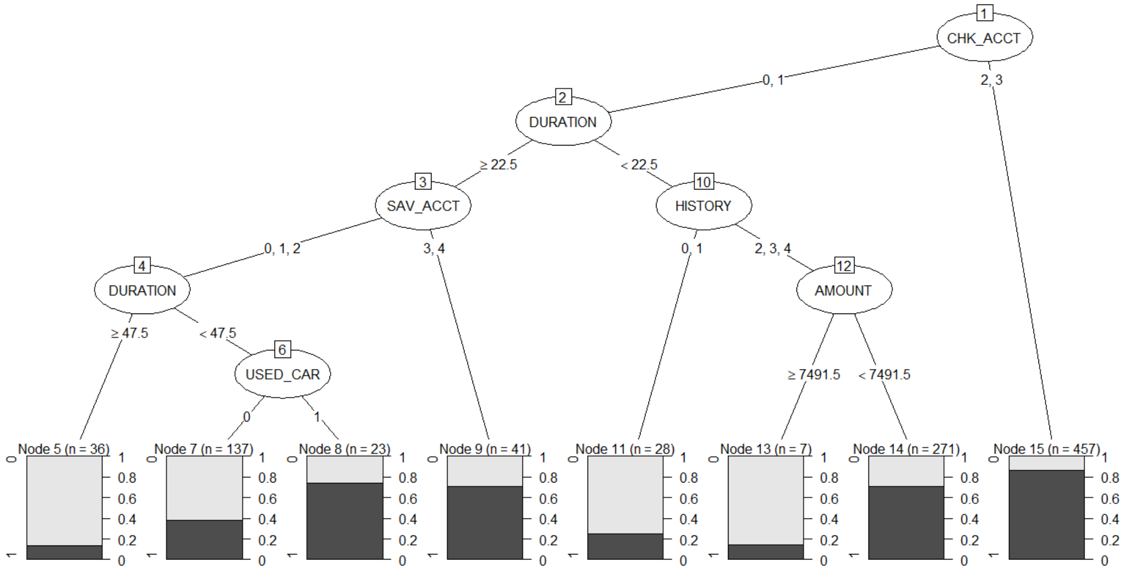
Step 5 - Pruning the tree to the least xerror

```
cp <- DT_Model2$cptable [opt,"CP"]
```

```
DT_Model_Pruned <- prune(DT_Model2, cp=cp)
```

```
plot(as.party(DT_Model_pruned))
```

You should get the following model. Try to explain the result using the cp table above.



Step 6 - Random Forest

Install the package for Random Forest

```
install.packages ("randomForest")
library(randomForest)
```

Run the Model

```
RF <- randomForest(RESPONSE~.,data=Data)
```

See the result

```
print(RF)
```

See importance of each predictor

```
importance(RF)
```

Plot the importance

```
varImpPlot(RF)
```

See the error vs. number of trees

```
plot(RF)
```