

# Tree Classification

W. Grossmann

```
# Tree Classifiers Demonstration
# -----
# Used Libraries
# -----
library("e1071")      # naiveBayes
## Warning: package 'e1071' was built under R version 3.2.5
library("klaR")      # NaiveBayes
## Warning: package 'klaR' was built under R version 3.2.5
## Loading required package: MASS
## Warning: package 'MASS' was built under R version 3.2.5
library("class")
library("MASS")
library("rpart")
library("tree")
## Warning: package 'tree' was built under R version 3.2.5
library(rattle)      # Fancy Visualization
## Warning: package 'rattle' was built under R version 3.2.5
## Rattle: Ein kostenloses grafisches Interface für Data Mining mit R.
## Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Geben Sie 'rattle()' ein, um Ihre Daten mischen.
library(rpart.plot) # Enhanced Tree Plots
## Warning: package 'rpart.plot' was built under R version 3.2.5
library(RColorBrewer) # Color selection
## Warning: package 'RColorBrewer' was built under R version 3.2.5
library(party)      # Alternative Decision Tree algorithms
## Warning: package 'party' was built under R version 3.2.5
## Loading required package: grid
## Loading required package: mvtnorm
## Warning: package 'mvtnorm' was built under R version 3.2.5
## Loading required package: modeltools
```

```

## Loading required package: stats4
## Loading required package: strucchange
## Warning: package 'strucchange' was built under R version 3.2.5
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 3.2.5
##
## Attaching package: 'zoo'
## Die folgenden Objekte sind maskiert von 'package:base':
##
##      as.Date, as.Date.numeric
## Loading required package: sandwich
library(partykit)      # Converts rpart objects into a binary
## Warning: package 'partykit' was built under R version 3.2.5
##
## Attaching package: 'partykit'
## Die folgenden Objekte sind maskiert von 'package:party':
##
##      cforest, ctree, ctree_control, edge_simple, mob, mob_control,
##      node_barplot, node_bivplot, node_boxplot, node_inner,
##      node_surv, node_terminal
library(caret)
## Warning: package 'caret' was built under R version 3.2.5
## Loading required package: lattice
## Loading required package: ggplot2
# -----
# Data
#-----

demo1<- read.csv("SimpleCRM1.csv",header=T,sep=";")
head(demo1)

##   DurCRM Sales User.Type UseS
## 1     10    12  private    1
## 2     24    36  business    1
## 3     28    48  business    1
## 4     45    20  private    1
## 5     30    34  private    1
## 6      3    21  private    1

```

```

attach(demo1)
use<-as.factor(UseS)

predictors<-demo1[,-4]
mod.nb <- naiveBayes(use ~ ., data = predictors, subset=1:11)
mod.nb

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##      0      1
## 0.4545455 0.5454545
##
## Conditional probabilities:
##   DurCRM
## Y   [,1]  [,2]
## 0 20.80000 14.35967
## 1 23.33333 15.01555
##
##   Sales
## Y   [,1]  [,2]
## 0 20.8 17.03526
## 1 28.5 13.17194
##
##   User.Type
## Y   business  private
## 0 0.6000000 0.4000000
## 1 0.3333333 0.6666667

predict(mod.nb,predictors)

## [1] 1 1 1 1 1 1 0 0 0 1 1 1
## Levels: 0 1

predict(mod.nb, demo1[,-4], type= "raw")

##      0      1
## [1,] 0.4642830 0.5357170
## [2,] 0.4832863 0.5167137
## [3,] 0.4841647 0.5158353
## [4,] 0.2539610 0.7460390
## [5,] 0.2269896 0.7730104
## [6,] 0.3554735 0.6445265
## [7,] 0.8190291 0.1809709
## [8,] 0.5647048 0.4352952
## [9,] 0.5326860 0.4673140
## [10,] 0.3913542 0.6086458

```

```

## [11,] 0.3561505 0.6438495
## [12,] 0.3092328 0.6907672

pred<-predict(mod.nb,predictors)

table(pred, demo1[,4])

##
## pred 0 1
##    0 3 0
##    1 2 6

#-----
#Prediction for new case
#-----

Duration_new<-8
Sales_new<-15
Type_new<-"private"
newcase<-cbind(Duration_new,Sales_new,Type_new)
test<-as.data.frame(newcase)
predict(mod.nb,test)

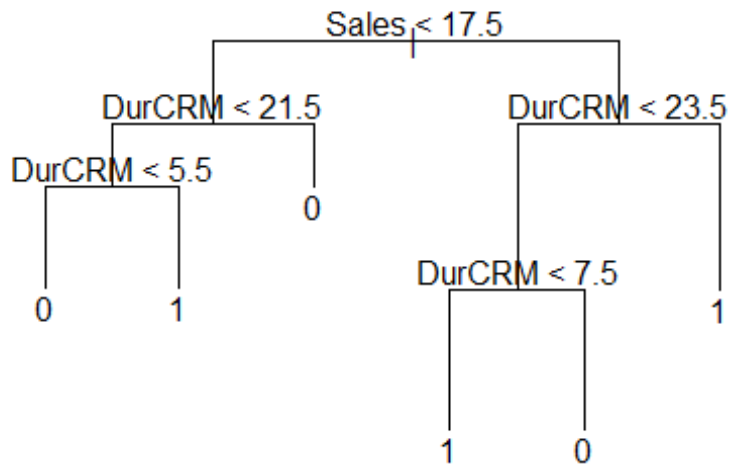
## [1] 1
## Levels: 0 1

#-----
# Tree Classifier with trees with deviance
#-----
mod.tr<- tree(use~., data=predictors, subset=1:11,
              mindev=1e-6, minsize=2)
mod.tr

## node), split, n, deviance, yval, (yprob)
##      * denotes terminal node
##
## 1) root 11 15.160 1 ( 0.4545 0.5455 )
##   2) Sales < 17.5 4 4.499 0 ( 0.7500 0.2500 )
##     4) DurCRM < 21.5 2 2.773 1 ( 0.5000 0.5000 )
##       8) DurCRM < 5.5 1 0.000 0 ( 1.0000 0.0000 ) *
##       9) DurCRM > 5.5 1 0.000 1 ( 0.0000 1.0000 ) *
##     5) DurCRM > 21.5 2 0.000 0 ( 1.0000 0.0000 ) *
##   3) Sales > 17.5 7 8.376 1 ( 0.2857 0.7143 )
##     6) DurCRM < 23.5 3 3.819 0 ( 0.6667 0.3333 )
##       12) DurCRM < 7.5 1 0.000 1 ( 0.0000 1.0000 ) *
##       13) DurCRM > 7.5 2 0.000 0 ( 1.0000 0.0000 ) *
##     7) DurCRM > 23.5 4 0.000 1 ( 0.0000 1.0000 ) *

plot(mod.tr)
text(mod.tr)

```



```
summary(mod.tr)
```

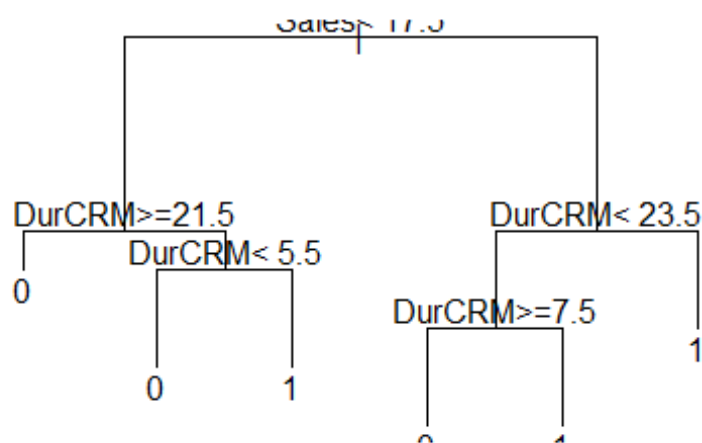
```
##
## Classification tree:
## tree(formula = use ~ ., data = predictors, subset = 1:11, mindev = 1e-06,
##       minsize = 2)
## Variables actually used in tree construction:
## [1] "Sales" "DurCRM"
## Number of terminal nodes: 6
## Residual mean deviance: 0 = 0 / 5
## Misclassification error rate: 0 = 0 / 11
```

```
predict(mod.tr, predictors)
```

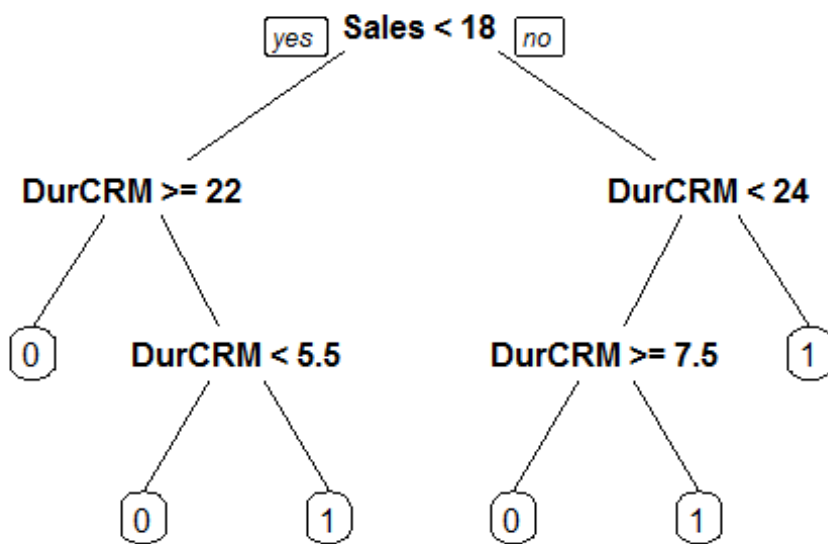
```
##    0 1
## 1  0 1
## 2  0 1
## 3  0 1
## 4  0 1
## 5  0 1
## 6  0 1
## 7  1 0
## 8  1 0
## 9  1 0
## 10 1 0
## 11 1 0
## 12 1 0
```

```
#=====
# Tree with rpart
#=====
modT.rpart<-rpart(use~., data = predictors,
                  control = rpart.control( minsplit = 2, cp = 0))

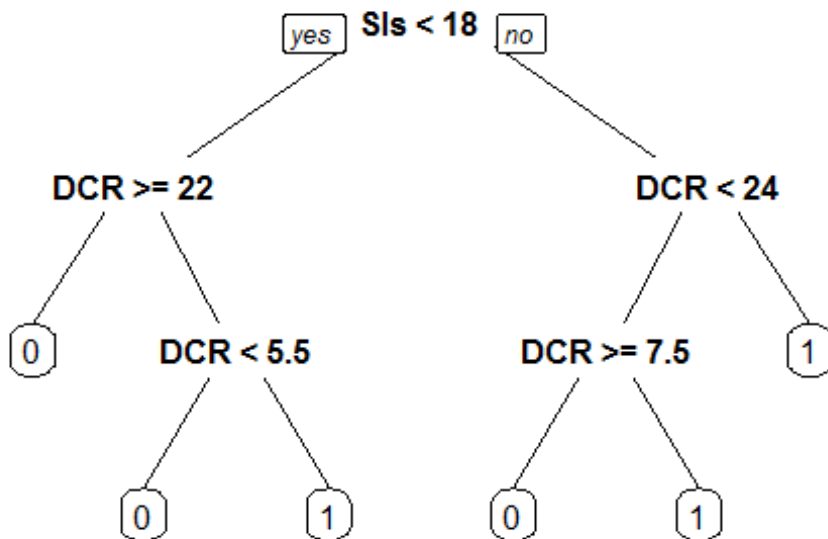
plot(modT.rpart)
text(modT.rpart)
```



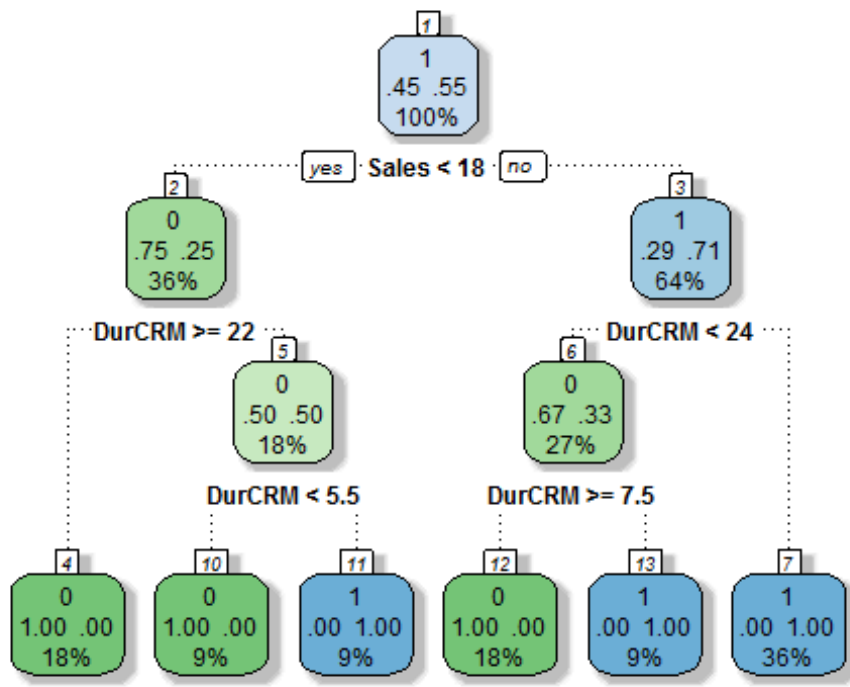
```
prp(modT.rpart)
```



```
prp(modT.rpart, varlen=3)
```



```
fancyRpartPlot(modT.rpart)
```



Rattle 2017-Apr-26 00:02:23 willi

```

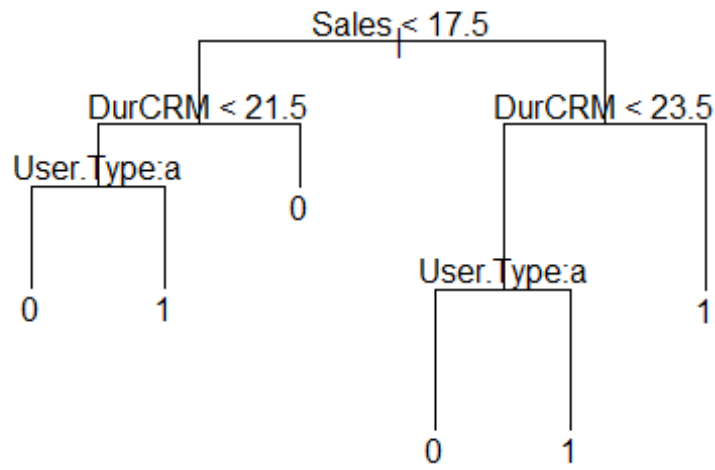
#-----
# Tree Classifier with Gini
#-----
mod.tr1<- tree(use~., data=predictors, split = "gini", subset=1:11,
               mindev=1e-6, minsize=2)
mod.tr1

## node), split, n, deviance, yval, (yprob)
## * denotes terminal node
##
## 1) root 11 15.160 1 ( 0.4545 0.5455 )
## 2) Sales < 17.5 4 4.499 0 ( 0.7500 0.2500 )
## 4) DurCRM < 21.5 2 2.773 1 ( 0.5000 0.5000 )
## 8) User.Type: business 1 0.000 0 ( 1.0000 0.0000 ) *
## 9) User.Type: private 1 0.000 1 ( 0.0000 1.0000 ) *
## 5) DurCRM > 21.5 2 0.000 0 ( 1.0000 0.0000 ) *
## 3) Sales > 17.5 7 8.376 1 ( 0.2857 0.7143 )
## 6) DurCRM < 23.5 3 3.819 0 ( 0.6667 0.3333 )
## 12) User.Type: business 2 0.000 0 ( 1.0000 0.0000 ) *
## 13) User.Type: private 1 0.000 1 ( 0.0000 1.0000 ) *
## 7) DurCRM > 23.5 4 0.000 1 ( 0.0000 1.0000 ) *

plot(mod.tr1)
text(mod.tr1)

```





```

summary(mod.tr1)

##
## Classification tree:
## tree(formula = use ~ ., data = predictors, subset = 1:11, split = "gini",
##       mindev = 1e-06, minsize = 2)
## Number of terminal nodes: 6
## Residual mean deviance: 0 = 0 / 5
## Misclassification error rate: 0 = 0 / 11

predict(mod.tr1, predictors[12,])

##      0 1
## 12  0 1

#-----
#Confusion Matrix
#-----
pred.tr<-predict(mod.tr1, type = "class")
pred.tr

## [1] 1 1 1 1 1 1 0 0 0 0 0
## Levels: 0 1

table(pred.tr, demo1[1:11,4])

##
## pred.tr 0 1
  
```

```
##      0 5 0
##      1 0 6

#-----
#simple tree
#-----
mod.tr2<- tree(use~., data=predictors,
               split = "gini", subset=1:11)

mod.tr2

## node), split, n, deviance, yval, (yprob)
##      * denotes terminal node
##
## 1) root 11 15.160 1 ( 0.4545 0.5455 )
##  2) DurCRM < 23.5 5 6.730 0 ( 0.6000 0.4000 ) *
##  3) DurCRM > 23.5 6 7.638 1 ( 0.3333 0.6667 ) *
```